Optimal prefix codes for pairs of geometrically-distributed random variables

Frédérique Bassino
LIPN UMR 7030
Université Paris 13 - CNRS, France
bassino@lipn.univ-paris13.fr

Julien Clément
GREYC UMR 6072
CNRS, Université de Caen, ENSICAEN, France
julien.clement@info.unicaen.fr

Gadiel Seroussi
HP Labs, Palo Alto, California and
Universidad de la República, Montevideo, Uruguay
gseroussi@ieee.org

Alfredo Viola
Universidad de la República, Montevideo, Uruguay, and
LIPN UMR 7030, Université Paris 13 - CNRS, France
viola@fing.edu.uy

Abstract

Optimal prefix codes are studied for pairs of independent, integer-valued symbols emitted by a source with a geometric probability distribution of parameter q, $0<q<1$. By encoding pairs of symbols, it is possible to reduce the redundancy penalty of symbol-by-symbol encoding, while preserving the simplicity of the encoding and decoding procedures typical of Golomb codes and their variants. It is shown that optimal codes for these so-called two-dimensional geometric distributions are *singular*, in the sense that a prefix code that is optimal for one value of the parameter q cannot be optimal for any other value of q. This is in sharp contrast to the one-dimensional case, where codes are optimal for positive-length intervals of the parameter q. Thus, in the two-dimensional case, it is infeasible to give a compact characterization of optimal codes for all values of the parameter q, as was done in the one-dimensional case. Instead, optimal codes are characterized for a discrete sequence of values of q that provide good coverage of the unit interval. Specifically, optimal prefix codes are described for $q = 2^{-1/k}$ ($k \geq 1$), covering the range $q \geq \frac{1}{2}$, and $q = 2^{-k}$ ($k > 1$), covering the range $q < \frac{1}{2}$. The described codes produce the expected reduction in redundancy with respect to the one-dimensional case, while maintaining low complexity coding operations.

This work was supported in part by ECOS project U08E02, and by PDT project 54/178 2006–2008.

I. Introduction

In 1966, Golomb [1] described optimal binary prefix codes for some geometric distributions over the nonnegative integers, namely, distributions with probabilities $p(i)$ of the form

$$p(i) = (1-q)q^i\,, \quad i \geq 0,$$

for some real-valued parameter q, $0 < q < 1$. In [2], these *Golomb codes* were shown to be optimal for *all* geometric distributions. These distributions occur, for example, when encoding *run lengths* (the original motivation in [1]), and in image compression when encoding prediction residuals, which are well-modeled by *two-sided geometric distributions*. Optimal codes for the latter were characterized in [3], based on some combinations and variants of Golomb codes. Codes based on the Golomb construction have the practical advantage of allowing the encoding of a symbol i using a simple explicit computation on the integer value of i, without recourse to nontrivial data structures or tables. This has led to their adoption in many practical applications (cf. [4],[5]).

Symbol-by-symbol encoding, however, can incur significant redundancy relative to the entropy of the distribution, even when dealing with sequences of independent, identically distributed random variables. One way to mitigate this problem, while keeping the simplicity and low latency of the encoding and decoding operations, is to consider short blocks of $d{>}1$ symbols, and use a prefix code for the blocks. In this paper, we study optimal prefix codes for pairs (blocks of length $d{=}2$) of independent, identically distributed geometric random variables, namely, distributions on pairs of nonnegative integers (i,j) with probabilities of the form

$$P(i,j) = p(i)p(j) = (1-q)^2 q^{i+j} \quad i,j \geq 0. \tag{1}$$

We refer to this distribution as a *two-dimensional geometric distribution (TDGD)*, defined on the alphabet of integer pairs $\mathcal{A} = \{\, (i,j) \mid i,j \geq 0 \,\}$. For succinctness, we denote a TDGD of parameter q by TDGD(q).

Aside from the mentioned practical motivation, the problem is of intrinsic combinatorial interest. It was proved in [6] (see also [7]) that, if the entropy[1] $-\sum_{i\geq 0} P(i) \log P(i)$ of a distribution over the nonnegative integers is finite, optimal codes exist and can be obtained, in the limit, from Huffman codes for truncated versions of the alphabet. However, the proof does not give a general way for effectively constructing optimal codes, and in fact, there are few families of distributions over countable alphabets for which an effective construction is known [8][9]. An algorithmic approach to building optimal codes is presented in [9], which covers geometric distributions and various generalizations. The approach, though, is not applicable to TDGDs, as explicitly noted in [9], and, to the best of our knowledge, no general constructions of optimal codes for TDGDs have been reported in the literature (except for the case $q = \frac{1}{2}$, which is trivial, cf. also [10]).

Some characteristic properties of the families of optimal codes for geometric and related distributions in the one-dimensional case turn out not to hold in the two-dimensional case. Specifically, the optimal

[1] $\log x$ and $\ln x$ will denote, respectively, the base-2 and the natural logarithm of x.

codes described in [1] and [3] correspond to binary trees of *bounded width*, namely, the number of codewords of any given length is upper-bounded by a quantity that depends only on the code parameters. Also, the family of optimal codes in each case partitions the parameter space into regions of positive volume, such that all the corresponding distributions in a region admit the same optimal code. These properties do not hold in the case of optimal codes for TDGDs. In particular, optimal codes for TDGDs turn out to be *singular*, in the sense that if a code $\mathcal{T}_q$ is optimal for TDGD(q), then $\mathcal{T}_q$ *is not* optimal for TDGD(q') for any parameter value $q' \neq q$. This result is presented in Section III. (A related but somewhat dual problem, namely, counting the number of distinct trees that can be optimal for a given source over a countable alphabet, is studied in [11].)

An important consequence of this singularity is that any set containing optimal codes for all values of q would be uncountable, and, thus, it would be infeasible to give a compact characterization of such a set, as was done in [1] or [3] for one-dimensional cases.[2] Thus, from a practical point of view, the best we can expect is to characterize optimal codes for countable sequences of parameter values. In this paper, we present such a characterization, for a sequence of parameter values that provides good coverage of the range of $0<q<1$. Specifically, in Section IV, we describe the construction of optimal codes for TDGD(q) with $q = 2^{-1/k}$ for integers $k \geq 1$,[3] covering the range $q \geq \frac{1}{2}$, and in Section V, we do so for TDGD(q) with $q = 2^{-k}$ for integers $k > 1$, covering the range $q < \frac{1}{2}$. In the case $q < \frac{1}{2}$, we observe that, as $k \to \infty$ ($q \to 0$), the optimal codes described converge to a *limit code*, in the sense that the codeword for any given pair (a, b) remains the same for all $k > k_0(a, b)$, where k_0 is a threshold that can be computed from a and b (this limit code is also mentioned in [10]). The codes in both constructions are of unbounded width. However, they are *regular* [12], in the sense that the corresponding infinite trees have only a finite number of non-isomorphic *whole subtrees* (i.e., subtrees consisting of a node and all of its descendants). This allows for deriving recursions and explicit expressions for the average code length, as well as feasible encoding/decoding procedures.

Practical considerations, and the redundancy of the new codes, are discussed in Section VI, where we present redundancy plots and comparisons with symbol-by-symbol Golomb coding and with the optimal code for a TDGD for each plotted value of q (optimal average code lengths for arbitrary values of q were estimated numerically to sufficiently high precision). We also derive an exact expression for the asymptotic oscillatory behavior of the redundancy of the new codes as $q \to 1$. The study confirms the redundancy gains over symbol-by-symbol encoding with Golomb codes, and the fact that the discrete sequence of codes presented provides a good approximation to the full class of optimal codes over the range of the parameter q.

Our constructions and proofs of optimality rely on the technique of Gallager and Van Voorhis [2], which was also used in [3]. As noted in [2], most of the work and ingenuity in applying the technique goes

[2] Loosely, by a compact characterization we mean one in which each code is characterized by a finite number of finite parameters, which drive the corresponding encoding/decoding procedures.

[3] These are the same distributions for which optimality of Golomb codes was originally established in [1].

into discovering appropriate "guesses" of the basic components on which the construction iterates, and in describing the structure of the resulting codes. With the correct guesses, the proofs are straightforward. The technique of [2] is reviewed in Section II, where we also introduce some definitions and notation that will be useful throughout the paper.

II. Preliminaries

A. Definitions

We are interested in encoding the alphabet $\mathcal{A}$ of integer pairs (i, j), $i, j \geq 0$, using a binary prefix code C. As usual, we associate C with a rooted (infinite) binary tree, whose leaves correspond, bijectively, to symbols in $\mathcal{A}$, and where each branch is labeled with a binary digit. The binary codeword assigned to a symbol is "read off" the labels on the path from the root to the corresponding leaf. The *depth* of a node x in a tree T, denoted $\text{depth}_T(x)$, is the number of branches on the path from the root to x. By extension, the depth (or *height*) of a finite tree is defined as the maximal depth of any of its nodes. A *level* of T is the set of all nodes at a given depth ℓ (we refer to this set as *level* ℓ). Let n_ℓ^T denote the number of leaves in level ℓ of T (we will sometimes omit the superscript T when clear from the context). We refer to the sequence $\{n_\ell^T\}_{\ell\geq 0}$ as the *profile* of T. Two trees will be considered *equivalent* if their profiles are identical. Thus, for a code C, we are only interested in its tree profile, or, equivalently, the *length distribution* of its codewords. Given the profile of a tree, and an ordering of $\mathcal{A}$ in decreasing probability order, it is always possible to define a canonical tree (say, by assigning leaves in alphabetical order) that uniquely defines a code for $\mathcal{A}$. Therefore, with a slight abuse of terminology, we will not distinguish between a code and its corresponding tree (or profile), and will refer to the same object sometimes as a tree and sometimes as a code. All trees considered in this paper are binary and, unless noted otherwise, *full*, i.e., every node in the tree is either a leaf or the parent of two children.[4] A tree is *balanced* (or *uniform*) if it has depth k, and 2^k leaves, for some $k \geq 0$. We denote such a tree by $\mathcal{U}_k$. We will restrict the use of the term *subtree* to refer to whole subtrees of T, i.e., subtrees that consist of a node and all of its descendants in T.

We call $s(i, j) = i + j$ the *signature* of $(i, j) \in \mathcal{A}$. For a given signature $s = s(i, j)$, there are $s+1$ pairs with signature s, all with the same probability, $P(s)=(1-q)^2 q^s$, under the distribution (1) on $\mathcal{A}$. Given a code C, symbols of the same signature can be freely permuted without affecting the properties of interest to us (e.g., average code length). Thus, for simplicity, we can also regard the correspondence between leaves and symbols as one between leaves and elements of the *multiset*

$$\hat{\mathcal{A}} = \{0, 1, 1, 2, 2, 2, \ldots, \underbrace{s, \ldots, s}_{s+1 \text{ times}}, \ldots\}. \tag{2}$$

[4] We use the usual "family" terminology for trees: nodes have children, parents, ancestors and descendants. We also use the common convention of visualizing trees with the root at the top and leaves at the bottom. Thus, ancestors are "up," and descendants are "down." Full trees are sometimes referred to in the literature as *complete*.

In constructing the tree, we do not distinguish between different occurrences of a signature s; for actual encoding, the $s{+}1$ leaves labeled with s are mapped to the symbols $(0,s),(1,s{-}1),\ldots,(s,0)$ in some fixed order. In the sequel, we will often ignore normalization factors for the signature probabilities $P(s)$ (in cases where normalization is inconsequential), and will use instead *weights* $w(s) = q^s$.

Consider a tree (or code) T for $\mathcal{A}$. Let U be a subtree of T, and let $s(x)$ denote the signature associated with a leaf x of U. Let $F(U)$ denote the set of leaves of U, referred to as its *fringe*. We define the *weight*, $w_q(U)$, of U as

$$w_q(U) = \sum_{x \in F(U)} q^{s(x)}\,,$$

and the *cost*, $\mathcal{L}_q(U)$, of U as

$$\mathcal{L}_q(U) = \sum_{x \in F(U)} \text{depth}_U(x) q^{s(x)}$$

(the subscript q may be omitted when clear from the context). When $U = T$, we have $w_q(T) = (1-q)^{-2}$, and $(1-q)^2\mathcal{L}_q(T)$ is the average code length of T. A tree T is *optimal* for TDGD(q) if $\mathcal{L}_q(T) \le \mathcal{L}_q(T')$ for any tree T'.

B. *Some basic objects and operations*

For $\alpha \ge 1$, we say that a finite source with probabilities $p_1 \ge p_2 \ge \cdots \ge p_N$, $N \ge 2$, is *α-uniform* if $p_1/p_N \le \alpha$. When $\alpha = 2$, the source is called *quasi-uniform*. An optimal code for a quasi-uniform source on N symbols consists of $2^{\lceil \log N \rceil}{-}N$ codewords of length $\lfloor \log N \rfloor$, and $2N{-}2^{\lceil \log N \rceil}$ codewords of length $\lceil \log N \rceil$, the shorter codewords corresponding to the more probable symbols [2]. We refer to such a code (or the associated tree) also as *quasi-uniform*, denote it by Q_N, and denote by $Q_N(i)$ the codeword it assigns to the symbol associated with p_i, $1{\le}i{\le}N$. For convenience, we define Q_1 as a null code, which assigns code length zero to the single symbol in the alphabet. Clearly, for integers $k \ge 0$, we have $Q_{2^k} = \mathcal{U}_k$. The *fringe thickness* of a finite tree T, denoted f_T, is the maximum difference between the depths of any two leaves of T. Quasi-uniform trees T have $f_T \le 1$, while uniform trees have $f_T = 0$. In Section IV we present a characterization of optimal codes of fringe thickness two for 4-uniform distributions, which generalizes the quasi-uniform case. This generalization will help in the characterization of the optimal codes for TDGD(q), $q = 2^{-1/k}$. The *concatenation* of two trees T and U, denoted $T \cdot U$, is obtained by attaching a copy of U to each leaf of T. Regarded as a code, $T \cdot U$ consists of all the possible concatenations $t \cdot u$ of a word $t \in T$ with one $u \in U$.

The *Golomb code* of order $k \ge 1$ [1], denoted G_k, encodes an integer i by concatenating $Q_k(i \bmod k)$ with a *unary* encoding of $\lfloor i/k \rfloor$ (e.g., $\lfloor i/k \rfloor$ ones followed by a zero). The first-order Golomb code G_1 is just the unary code, whose corresponding tree consists of a root with one leaf child on the branch labeled '0', and, recursively, a copy of G_1 attached to the child on the branch labeled '1'. Thus, we have $G_k = Q_k \cdot G_1$.

C. The Gallager-Van Voorhis method

When proving optimality of infinite codes for TDGDs, we will rely on the method due to Gallager and Van Voorhis [2], which is briefly outlined below, adapted to our terminology.

- Define a sequence of finite *reduced alphabets* $(\mathcal{S}_s)_{s=-1}^{\infty}$. The reduced alphabet $\mathcal{S}_s$ is a multiset containing the signatures $0, 1, \ldots, s$ (with multiplicities as in (2)), and a finite number of (possibly infinite) subsets of $\hat{\mathcal{A}}$, referred to as *virtual symbols*, which form a partition of the multiset of signatures strictly greater than s. We naturally associate with each virtual symbol a weight equal to the sum of the weights of the signatures it contains.
- Verify that the sequence $(\mathcal{S}_s)_{s=-1}^{\infty}$ is compatible with the bottom-up Huffman procedure. This means that after a number of merging steps of the Huffman algorithm on the reduced alphabet $\mathcal{S}_s$, one gets $\mathcal{S}_{s'}$ with $s' < s$. Proceed recursively, until $s' = -1$.
- Apply the Huffman algorithm to $\mathcal{S}_{-1}$.

While the sequence of reduced alphabets $\mathcal{S}_s$ can be seen as evolving "bottom-up," the infinite code C constructed results from a "top-down" sequence of corresponding finite codes C_s, which grow with s and unfold by recursive reversal of the mergers in the Huffman procedure. One shows that the sequence of codes $(C_s)_{s\geq -1}$ *converges* to the infinite code C, in the sense that for every $i \geq 1$, with codewords of C_s consistently sorted, the ith codeword of C_s is eventually constant when s grows, and equal to the ith codeword of C. A corresponding convergence argument on the sequence of average code lengths then establishes the optimality of C.

This method was successfully applied to characterize infinite optimal codes in [2] and [3]. While the technique is straightforward once appropriate reduced alphabets are defined, the difficulty in each case is to guess the structure of these alphabets. In a sense, this is a self-bootstrapping procedure, where one needs to guess the structure of the codes sought, and use that structure to define the reduced alphabets, which, in turn, serve to prove that the guess was correct. We will apply the Gallager-Van Voorhis method to prove optimality of codes for certain families of TDGDs in Sections IV and V. In each case, we will emphasize the definition and structure of the reduced alphabets, and show that they are compatible with the Huffman procedure. We will omit the discussion on convergence, and formal induction proofs, since the arguments are essentially the same as those of [2] and [3].

III. Singularity of optimal codes for TDGDs

In the case of one-dimensional geometric distributions, the unit interval $(0, 1)$ is partitioned into an infinite sequence of semi-open intervals $(q_{m-1}, q_m]$, $m \geq 1$, such that the Golomb code G_m is optimal for all values of the distribution parameter q in the interval $q_{m-1} < q \leq q_m$. Specifically, for $m \geq 0$, q_m is the (unique) nonnegative root of the equation $q^m + q^{m+1} - 1 = 0$ [2]. Thus, we have $q_0 = 0$, $q_1 = (\sqrt{5} - 1)/2 \approx 0.618, q_2 \approx 0.755$, etc. A similar property holds in the case of two-sided geometric distributions [3], where the two-dimensional parameter space is partitioned into a countable sequence of patches such that all the distributions with parameter values in a given patch admit the same optimal code. In this section, we prove that, in sharp contrast to these examples, optimal codes for TDGDs are *singular*, in the sense that a code that is optimal for a certain value of the parameter q *cannot* be optimal for any other value of q. More formally, we present the following result.

Theorem 1: Let q and q_1 be real numbers in the interval $(0, 1)$, with $q \neq q_1$, and let $\mathcal{T}_q$ be an optimal tree for TDGD(q). Then, $\mathcal{T}_q$ is not optimal for TDGD(q_1).

We will prove Theorem 1 through a series of lemmas, which will shed more light on the structure of optimal trees for TDGDs.

Lemma 1: Leaves with a given signature s in $\mathcal{T}_q$ are found in at most two consecutive levels of $\mathcal{T}_q$.

Proof: Let d_0 and d_1 denote, respectively, the minimum and maximum depths of a leaf with signature s in $\mathcal{T}_q$. Assume, contrary to the claim of the lemma, that $d_1 > d_0 + 1$. We transform $\mathcal{T}_q$ into a tree $\mathcal{T}'_q$ as follows. Pick a leaf with signature s at level d_0, and one at level d_1. Place both signatures s as children of the leaf at level d_0, which becomes an internal node. Pick any signature s' from a level strictly deeper than d_1, and move it to the vacant leaf at level d_1. Tracking changes in the code lengths corresponding to the affected signatures, and their effect on the cost, we have

$$\mathcal{L}_q(\mathcal{T}'_q) = \mathcal{L}_q(\mathcal{T}_q) + q^s(d_0 - d_1 + 2) - q^{s'}\delta, \tag{3}$$

where δ is a positive integer. By our assumption, the quantity multiplying q^s in (3) is non-positive, and we have $\mathcal{L}_q(\mathcal{T}'_q) < \mathcal{L}_q(\mathcal{T}_q)$, contradicting the optimality of $\mathcal{T}_q$. Therefore, we must have $d_1 \leq d_0 + 1$. ∎

A *gap* in a tree T is a non-empty set of consecutive levels containing only internal nodes of T, and such that both the level immediately above the set and the level immediately below it contain at least one leaf each. The corresponding *gap size* is defined as the number of levels in the gap. It follows immediately from Lemma 1 that in an optimal tree, if the largest signature above a gap is s, then the smallest signature below the gap is $s + 1$.

Lemma 2: Let $\mathcal{T}_q$ be an optimal tree for TDGD(q), and let $k = 1 + \lfloor \log q^{-1} \rfloor$. Then, for all sufficiently large s, the size g of any gap between leaves of signature s and leaves of signature $s + 1$ in $\mathcal{T}_q$ satisfies $g \leq k - 1$.

Proof: Assume that an optimal tree $\mathcal{T}_q$ is given.

Case $q > \frac{1}{2}$. In this case, $k = 1$, and the claim of the lemma means that there can be *no gaps* in the tree from a certain level on. Assume that there is a gap between level d with signatures s, and level d' with signatures $s + 1$, $d' - d \geq 2$. By Lemma 1, all signatures $s + 1$ are either in level d' or in level $d' + 1$. By rearranging nodes within levels, we can assume that there is a subtree of $\mathcal{T}_q$ of height at most two, rooted at a node v of depth $d' - 1 \geq d + 1$, and containing at least two leaves of signature $s + 1$. Hence, the weight of the subtree satisfies

$$w(v) \geq 2q^{s+1} > q^s\,, \tag{4}$$

and switching a leaf s on level d with node v on level $d' - 1$ decreases the cost of $\mathcal{T}_q$, in contradiction with its optimality (when switching nodes, we carry also any subtrees rooted at them). Therefore, there can be no gap between the level containing signatures s and $s + 1$, as claimed. Notice that this holds for all values of s, regardless of level.

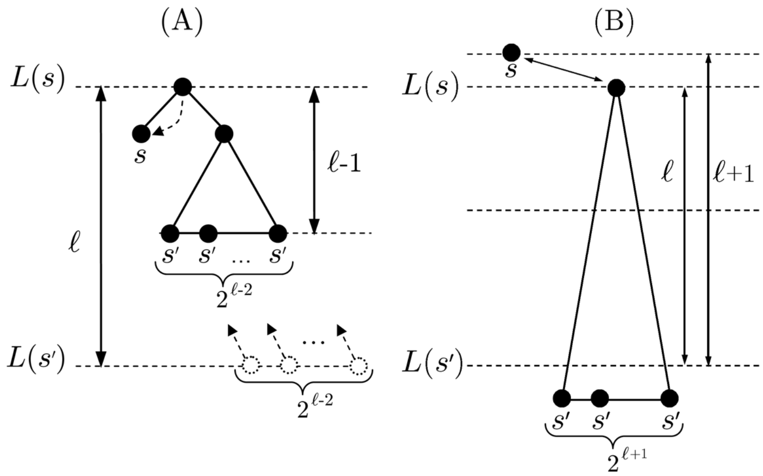


Fig. 1. Tree transformations.

Case $q = \frac{1}{2}$. In this case, the TDGD is dyadic, the optimal profile is uniquely determined, and it and has no gaps (the optimal profile is that of $G_1 \cdot G_1$).
Case $q < \frac{1}{2}$. Assume that $s \geq 2^k - 2$, and that there is a gap of size g between signatures s at level d, and signatures $s+1$ at level $d+g+1$. Signatures $s+1$ may also be found at level $d+g+2$. By a rearrangement of nodes that preserves optimality, and by our assumption on s, we can assume that there is a subtree of $\mathcal{T}_q$ rooted at a node v at level $d+g+1-k$, and containing at least 2^k leaves with signature $s+1$, including some at level $d+g+1$. Thus, we have

$$w(v) \geq 2^k q^{s+1} > q^s = w(s), \tag{5}$$

the second inequality following from the definition of k. Therefore, we must have $d+g+1-k \leq d$, or equivalently, $g \leq k-1$, for otherwise exchanging v and s would decrease the cost, contradicting the optimality of $\mathcal{T}_q$. ■

Next, we bound the rate of change of signature magnitudes as a function of depth in an optimal tree. Together with the bound on gap sizes in Lemma 2, this will lead to the proof of Theorem 1. It follows from Lemma 1 that for every signature $s \geq 0$ there is a level of $\mathcal{T}_q$ containing at least one half of the $s+1$ leaves with signature s. We denote the depth of this level by $L(s)$ (with some fixed policy for ties).

Lemma 3: Let s be a signature, and $\ell \geq 2$ a positive integer such that $s \geq 2^{\ell+2} - 1$, and such that $L(s') = L(s) + \ell$ for some signature $s' > s$. Then, in an optimal tree $\mathcal{T}_q$ for TDGD(q), we have

$$\frac{\ell - 2}{\log q^{-1}} \leq \; s' - s \; \leq \frac{\ell + 1}{\log q^{-1}} \; . \tag{6}$$

Proof: Since $s' > s \geq 2^{\ell+2} - 1 > 2^{\ell-1} - 1$, by the definition of $L(s')$, there are more than $2^{\ell-2}$ leaves with signature s' at level $L(s')$. We perform the following transformation (depicted in Figure 1(A)) on the tree $\mathcal{T}_q$, yielding a modified tree $\mathcal{T}'_q$: choose a leaf with signature s at level $L(s)$, and graft to it a tree with a left subtree consisting of a leaf with signature s ("moved" from the root of the subtree), and a right subtree that is a balanced tree of height $\ell - 2$ with $2^{\ell-2}$ leaves of signature s'. These signatures come from $2^{\ell-2}$ leaves at level $L(s')$ of $\mathcal{T}_q$, which are removed. It is easy to verify that the modified tree $\mathcal{T}_q'$ defines a valid, albeit incomplete, code for the alphabet of a TDGD. Next, we estimate the change, Δ, in cost due to this transformation. We have

$$\Delta = \mathcal{L}_q(\mathcal{T}_q') - \mathcal{L}_q(\mathcal{T}_q) \leq q^s - 2^{\ell-2}q^{s'} .$$

The term q^s is due to the increase, by one, in the code length for the signature s, which causes an increase in cost, while the term $-2^{\ell-2}q^{s'}$ is due to the decrease in code length for $2^{\ell-2}$ signatures s', which produces a decrease in cost. Since $\mathcal{T}_q$ is optimal, we must have $\Delta \geq 0$, namely,

$$0 \leq q^s - 2^{\ell-2}q^{s'} = q^s\left(1 - 2^{\ell-2}q^{s'-s}\right),$$

and thus, $2^{\ell-2}q^{s'-s} \leq 1$, from which the lower bound in (6) follows. (Note: clearly, the condition $s \geq 2^{\ell-1} - 1$ would have sufficed to prove the lower bound; the stricter condition of the lemma will be required for the upper bound, and was adopted here for uniformity.)

To prove the upper bound, we apply a different modification to $\mathcal{T}_q$. Here, we locate $2^{\ell+1}$ signatures s' at level $L(s')$, and rearrange the level so that these signatures are the leaves of a balanced tree of height $\ell + 1$, rooted at depth $L(s) - 1$. The availability of the required number of leaves at level $L(s')$ is guaranteed by the conditions of the lemma. We then exchange the root of this subtree with a leaf of signature s at level $L(s)$. The situation, after the transformation, is depicted in Figure 1(B). The resulting change in cost is computed as follows.

$$\Delta = \mathcal{L}_q(\mathcal{T}_q') - \mathcal{L}_q(\mathcal{T}_q) \leq -q^s + 2^{\ell+1}q^{s'} .$$

As before, we must have $\Delta \geq 0$, from which the upper bound follows. ■

We are now ready to prove Theorem 1.

Proof of Theorem 1: We assume, without loss of generality, that $q_1 > q$, and we write $q_1 = q(1+\varepsilon)$, $0 < \varepsilon < q^{-1} - 1$. In $\mathcal{T}_q$, choose a sufficiently large signature s (the meaning of "sufficiently large" will be specified in the sequel), and a node of signature s at level $L(s)$. Let $s' > s$ be a signature such that $\ell \triangleq L(s') - L(s) \geq 2$. We apply the transformation of Figure 1(A) to $\mathcal{T}_q$, yielding a modified tree $\mathcal{T}_q'$. We claim that when weights are taken with respect to TDGD(q_1), and with an appropriate choice of the parameter ℓ, $\mathcal{T}_q'$ will have strictly lower cost than $\mathcal{T}_q$. Therefore, $\mathcal{T}_q$ is not optimal for TDGD(q_1). To prove the claim, we compare the costs of $\mathcal{T}_q$ and $\mathcal{T}_q'$ with respect to TDGD(q_1). Reasoning as in the proof of the lower bound in Lemma 3, we write

$$\begin{aligned}\Delta &= \mathcal{L}_{q_1}(\mathcal{T}_q') - \mathcal{L}_{q_1}(\mathcal{T}_q) \leq q_1^s - 2^{\ell-2}q_1^{s'} \\ &= q_1^s\left(1 - 2^{\ell-2}q_1^{s'-s}\right) \leq q_1^s\left(1 - 2^{\ell-2}q_1^{\frac{\ell+1}{\log q^{-1}}}\right),\end{aligned} \tag{7}$$

where the last inequality follows from the upper bound in Lemma 3. It follows from (7) that we can make Δ negative if

$$\ell - 2 + \frac{\ell+1}{\log q^{-1}} \log q_1 > 0.$$

Writing q_1 in terms of q and ε, and after some algebraic manipulations, the above condition is equivalent to

$$\ell > 3\frac{\log q^{-1}}{\log(1+\varepsilon)} - 1\,. \tag{8}$$

Hence, choosing a large enough value of ℓ, we get $\Delta < 0$, and we conclude that the tree $\mathcal{T}_q$ is not optimal for TDGD(q_1), subject to an appropriate choice of s, which we discuss next.

The argument above relies strongly on Lemma 3. We recall that in order for this lemma to hold, ℓ and the signature s must satisfy the condition $s \geq 2^{\ell+2} - 1$. Now, it could happen that, after choosing ℓ according to (8) and then s according to the condition of Lemma 3, the level $L(s) + \ell$ does not contain $2^{\ell-2}$ signatures s' as required (e.g., when the level is part of a gap). This would force us to increase ℓ, which could then make s violate the condition of the lemma. We would then need to increase s, and re-check ℓ, in a potentially vicious circle. The bound on gap sizes of Lemma 2 allows us to avoid this trap. The bound in the lemma depends only on q and thus, for a given TDGD, it is a constant, say g_q. Thus, first, we choose a value ℓ_0 satisfying the constraint on ℓ in (8). Then, we choose $s \geq 2^{\ell_0+g_q+4}$. Now, we try $\ell = \ell_0, \ell_0+1, \ell_0+2, \ldots,$ in succession, and check whether level $L(s)+\ell$ contains enough of the required signatures. By Lemmas 1 and 2, an appropriate level $L(s')$ will be found for some $\ell \leq \ell_0 + g_q + 2$. For such a value of ℓ, we have $2^{\ell+2} - 1 \leq 2^{\ell_0+g_q+4} - 1 < s$, satisfying the condition of Lemma 3. This condition, in turn, guarantees also that there are at least $2^{\ell-2}$ signatures s' at $L(s')$, as required. ■

IV. Optimal codes for TDGDs with parameters $q = 2^{-1/k}$

It follows from the results of Section III that it is infeasible to provide a compact description of optimal codes for TDGDs covering all values of the parameter q, as can be done with one-dimensional geometric distributions [1], [2] or their two-sided variants [3]. Instead, we describe optimal prefix codes for a discrete sequence of values of q, which provide good coverage of the parameter range. In this section, we study optimal codes for TDGDs with parameters $q = 2^{-1/k}$ for integers $k \geq 1$, i.e., $q \geq \frac{1}{2}$, while in Section V we consider parameters of the form $q = 2^{-k}$, $k > 1$, covering the range $q < \frac{1}{2}$ (the two parameter sequences coincide at $k = 1$, $q = \frac{1}{2}$, which we choose to assign to the case covered in this section).

A. Initial characterization of optimal codes for $q = 2^{-1/k}$

The following theorem characterizes optimal codes for TDGDs of parameter $q = 2^{-1/k}$, $k \geq 1$, in terms of unary codes and Huffman codes for certain finite distributions. In Subsection IV-C we further refine the characterization by providing explicit descriptions of these Huffman codes.

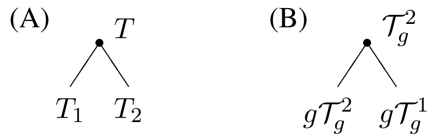


Fig. 2. Graphical representations for trees with associated weights.

Theorem 2: An optimal prefix code C_k for TDGD(q), with $q = 2^{-1/k}$, $k \geq 1$, is given by

$$C_k(i,j) = T_k(i \bmod k, j \bmod k) \cdot G_1(\lfloor \tfrac{i}{k} \rfloor) \cdot G_1(\left\lfloor \tfrac{j}{k} \right\rfloor), \tag{9}$$

where G_1 is the unary code, and T_k, referred to as the *top code*, is an optimal code for the finite source

$$\hat{\mathcal{A}}_k = \{(i,j) \mid 0 \leq i,j < k\} \text{ with weights } w(i,j) = q^{i+j} \,. \tag{10}$$

Remarks.

1) Theorem 2 can readily be generalized to blocks of $d > 2$ symbols. For simplicity, we present the proof for $d = 2$.
2) Notice that $C_k(i,j)$ concatenates the "unary" parts of the codewords for i and j in a Golomb code of order k (as if encoding i and j separately), but encodes the "binary" part jointly by means of T_k, which, in general, does not yield the concatenation of the respective "binary" parts $Q_k(i)$ and $Q_k(j)$. However, when $k = 1$ and $k = 2$, C_k *is* equivalent to the full concatenation $G_k \cdot G_k$. When $k = 1$, the code T_k is void, and $C_1 = G_1 \cdot G_1$. The parameter in this case is $q = \frac{1}{2}$, the geometric distribution is dyadic, and the code redundancy is zero. When $k = 2$, we have $q = 1/\sqrt{2}$ and the finite source $\hat{\mathcal{A}}_k$ has four symbols with respective weights $\{\,1,\, 1/\sqrt{2},\, 1/\sqrt{2}, 1/2\,\}$. This source is quasi-uniform, and, therefore, it admits Q_4 as an optimal tree. This is a balanced tree of depth two, which can also be written as $Q_4 = Q_2 \cdot Q_2$. Thus, we have $C_2 = G_2 \cdot G_2$. Later on in the section, in Corollary 1, we will prove that this situation will not repeat for larger values of k: the "symbol by symbol" code $G_k \cdot G_k$ is strictly suboptimal for TDGD($2^{-1/k}$) for $k > 2$.

In deriving the proof of Theorem 2 and in subsequent sections, we shall make use of the following notations to describe and operate on some infinite trees with weights associated to their leaves. We denote by $\boxed{v}$ the trivial tree consisting of a single node (leaf) of weight v. Given a tree T and a scalar g, gT denotes the tree T with all its weights multiplied by g. Given trees T_1 and T_2, the graphic notation in Figure 2(A) represents a tree T consisting of a root node with T_1 as its left subtree and T_2 as its right subtree, each contributing its respective multiset of leaf weights. As a multiset of weights, T is the (multiset) union of T_1 and T_2. We will also use the notation $[\,T_1\ \ T_2\,]$ to represent the same multiset but with a different graph structure, namely, the forest consisting of the separate trees T_1 and T_2. We denote by $\mathcal{T}_g^1$ the tree of a unary code whose leaf at each depth $i \geq 1$ has weight g^i, and by $\mathcal{T}_g^2$ the structure in Figure 2(B). It is readily verified that $\mathcal{T}_g^2$ corresponds to the concatenation of two unary codes, with each of the $i-1$ leaves at depth $i \geq 2$ of $\mathcal{T}_g^2$ carrying weight g^i. In particular, as shown in Figure 3,

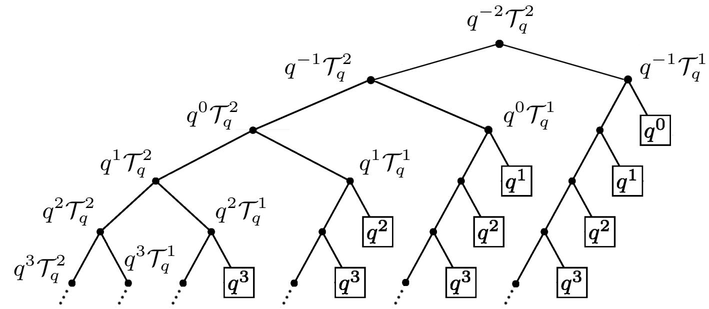


Fig. 3. The tree $q^{-2}\mathcal{T}_q^2$.

the tree $q^{-2}\mathcal{T}_q^2$ corresponds to the optimal tree for the dyadic TDGD with $q = \frac{1}{2}$, where each leaf is weighted according to the signature of the symbol it encodes.

The following lemma follows directly from the above definitions, applying elementary symbolic manipulations on geometric sums.

Lemma 4: For any real number g, $0 < g < 1$, we have $w(\mathcal{T}_g^2) = w(\mathcal{T}_g^1)^2 = \left(\frac{g}{1-g}\right)^2$. In particular, if $q = 2^{-1/k}$, we have $w(\mathcal{T}_{q^k}^2) = w(\mathcal{T}_{q^k}^1) = 1$.

We rely on this observation in the construction and proof of Theorem 2 below. In the proof, when defining virtual symbols, we further overload notation and regard trees of the form $q^r\mathcal{T}_{q^k}^d$ also as multisets of signatures, with a signature s for each leaf of the tree with weight q^s.

Proof of Theorem 2: We use the Gallager-Van Voorhis construction [2]. For $s \geq 0$, define the reduced alphabet

$$\mathcal{R}_s = \mathcal{H}_s \cup \mathcal{F}_s,$$

where

$$\mathcal{H}_s = \{i \in \hat{\mathcal{A}} \mid i < s\}$$

and

$$\mathcal{F}_s = \bigcup_{i=0}^{k-1} \{\underbrace{q^{s+i}\mathcal{T}_{q^k}^2,}_{k \text{ times}} \underbrace{q^{s+i}\mathcal{T}_{q^k}^1,}_{\substack{s+k+i+1\\ \text{times}}} \underbrace{s+i}_{\substack{s+i+1\\ \text{times}}}\}.$$

The multisets $q^{s+i}\mathcal{T}_{q^k}^1$ and $q^{s+i}\mathcal{T}_{q^k}^2$ play the role of virtual symbols in the reduced alphabets (we omit the qualifier 'virtual' in the sequel). It is readily verified that all the weights in $\mathcal{F}_s$ are smaller than the weights in $\mathcal{H}_s$. Since $q = 2^{-1/k}$, by Lemma 4, we have $w(q^{s+i}\mathcal{T}_{q^k}^2) = w(q^{s+i}\mathcal{T}_{q^k}^1) = w(s+i)$. Thus, we can apply steps of the Huffman procedure to $\mathcal{F}_s$ in such way that the $s+i+1$ signatures $s+i$ are merged with $s+i+1$ symbols $q^{s+i}\mathcal{T}_{q^k}^1$, resulting in $s+i+1$ trees $q^{s+i-k}\mathcal{T}_{q^k}^1$. The remaining k symbols $q^{s+i}\mathcal{T}_{q^k}^1$ can be merged with the k symbols $q^{s+i}\mathcal{T}_{q^k}^2$, resulting in k trees $q^{s+i-k}\mathcal{T}_{q^k}^2$ when i ranges from $k-1$ down to 0. After this sequence of Huffman mergings, $\mathcal{R}_s$ is transformed into $\mathcal{R}_{s-k}$, as long as

$s \geq 0$. Starting from $s = tk$ for some $t > 0$, the procedure eventually leads to

$$\mathcal{R}_{-k} = \cup_{i=0}^{k-1} \{ \underbrace{q^{i-k}\mathcal{T}^2_{q^k}}_{\substack{k \\ \text{times}}}, \underbrace{q^{i-k}\mathcal{T}^1_{q^k}}_{\substack{i+1 \\ \text{times}}} \} .$$

Formally, our reduced source $\mathcal{R}_{tk},\ t \in \mathbb{Z}$, corresponds to $\mathcal{S}_t$ in our description of the Gallager-Van Voorhis construction in Section II-C. Thus, the iteration leads to $\mathcal{S}_{-1}$, as called for in the construction. Now, each symbol $q^{i-k}\mathcal{T}^1_{q^k}$ in $\mathcal{S}_{-1}$ can be merged with a symbol $q^{i-k}\mathcal{T}^2_{q^k}$, leading, by the definition of $\mathcal{T}^2_g$ (see Figure 2(B)), to a reduced source

$$\mathcal{S}_T = \{ \underbrace{q^{-2k}\mathcal{T}^2_{q^k}}_{\substack{1 \\ \text{time}}}, \underbrace{q^{-2k+1}\mathcal{T}^2_{q^k}}_{\substack{2 \\ \text{times}}}, \underbrace{q^{-2k+2}\mathcal{T}^2_{q^k}}_{\substack{3 \\ \text{times}}}, \ldots, \underbrace{q^{-k-1}\mathcal{T}^2_{q^k}}_{\substack{k \\ \text{times}}}, \underbrace{q^{-k}\mathcal{T}^2_{q^k}}_{\substack{k-1 \\ \text{times}}}, \ldots, \underbrace{q^{-3}\mathcal{T}^2_{q^k}}_{\substack{2 \\ \text{times}}}, \underbrace{q^{-2}\mathcal{T}^2_{q^k}}_{\substack{1 \\ \text{time}}} \}.$$

We now take a common "factor" $q^{-2k}\mathcal{T}^2_{q^k}$ from each symbol of $\mathcal{S}_T$. By the discussion of Figures 2 and 3, this factor corresponds to a copy of $G_1 \cdot G_1$, with weights that get multiplied by q^k every time the depth increases by 1. After the common factor is taken out, the source $\mathcal{S}_T$ becomes the source $\hat{\mathcal{A}}_k$ of (10), to which the Huffman procedure needs to be applied to complete the code construction. Thus, the optimal code has the claimed structure. ∎

To make the result of Theorem 2 completely explicit, it remains to characterize an optimal prefix code for the finite source $\hat{\mathcal{A}}_k$ of (10). The following lemma presents some basic properties of $\hat{\mathcal{A}}_k$ and its optimal trees. Recall the definitions of α-uniformity and fringe thickness from Section II.

Lemma 5: The source $\hat{\mathcal{A}}_k$ is 4-uniform, and it has an optimal tree T of fringe thickness $f_T \leq 2$.

Proof: It follows from (10) and the relation $q^k = \frac{1}{2}$ that the maximal ratio between weights of symbols in $\hat{\mathcal{A}}_k$ is $q^{-2k+2} = 4q^2 < 4$. Hence, $\hat{\mathcal{A}}_k$ is 4-uniform. The claim on the optimal tree holds trivially for $k \leq 2$, in which case the optimal tree for $\hat{\mathcal{A}}_k$ is uniform. To prove the claim for $k > 2$, consider the multiset $\hat{\mathcal{A}}^*_k \subseteq \hat{\mathcal{A}}_k$ consisting of the lightest $2\lceil \frac{k(k-1)}{4} \rceil$ signatures in $\hat{\mathcal{A}}_k$, i.e.,

$$\hat{\mathcal{A}}^*_k = K \cup \{ \underbrace{k, k, \ldots, k}_{k-1 \text{ times}}, \underbrace{k+1, \ldots, k+1}_{k-2 \text{ times}}, \ldots,\ 2k-3, 2k-3,\ 2k-2 \} ,$$

where $K = \{k-1\}$ if $k \bmod 4 \in \{2, 3\}$, or K is empty otherwise. The sum of the two smallest weights of signatures in $\hat{\mathcal{A}}^*_k$ satisfies

$$w(2k-2) + w(2k-3) = q^{2k-2} + q^{2k-3} = q^{2k-2}(1 + q^{-1}) = \frac{1}{2}(1 + q^{-1})q^{k-2} > w(k-2) .$$

The sum of the two largest weights in $\hat{\mathcal{A}}^*_k$, on the other hand, is either q^0 if $k \bmod 4 \in \{0, 1\}$, or $\frac{1}{2}(1 + q^{-1})$ otherwise. Therefore, if the Huffman procedure is applied to $\hat{\mathcal{A}}_k$, every pair of consecutive elements of $\hat{\mathcal{A}}^*_k$ will be merged, without involving a previously merged pair. The ratio of the largest to the smallest weight remaining after these mergings is at most $\frac{1}{2}(1+q^{-1})/q^{k-1} = q+1 < 2$. Hence, the resulting source is quasi-uniform and has a quasi-uniform optimal tree. Therefore, completing the Huffman procedure for $\hat{\mathcal{A}}_k$ results in an optimal tree of fringe thickness at most two. ∎

To complete the explicit description of an optimal tree for $\hat{\mathcal{A}}_k$, we will rely on a characterization of trees T with $f_T \leq 2$ that are optimal for 4-uniform sources.[5] This characterization is presented next.

B. *Optimal trees with $f_T \leq 2$ for 4-uniform sources*

To proceed as directly as possible to the construction of an optimal tree for $\hat{\mathcal{A}}_k$, we defer the proofs of results in this subsection to Appendix A. We start by characterizing all the possible profiles for a tree T with N leaves, and $f_T \leq 2$. Let T be such a tree, let $m = \lceil \log N \rceil$, and denote by n_ℓ the number of leaves at depth ℓ in T.

Lemma 6: The profile of T satisfies $n_\ell = 0$ for $\ell < m-2$ and $\ell > m+1$, and either $n_{m-2} = 0$ or $n_{m+1} = 0$ (or both, when $f_T \leq 1$).

It follows from Lemma 6 that T is fully characterized by the quadruple $(n_{m-2}, n_{m-1}, n_m, n_{m+1})$, with either $n_{m-2} = 0$ or $n_{m+1} = 0$. We say T is *long* if $n_{m-2} = 0$, and that T is *short* if $n_{m+1} = 0$. Defining $M = m - \sigma$, where $\sigma = 1$ if T is short, or 0 if it is long, a tree with $f_T \leq 2$ can be characterized more compactly by a triple of nonnegative integers $\mathbf{N}_T = (n_{M-1}, n_M, n_{M+1})$. We will also refer to this triple as the *(compact) profile* of T, with the associated parameters N, m, and σ understood from the context. Notice that when $n_{m-2} = n_{m+1} = 0$, T is the quasi-uniform tree Q_N, and (abusing the metaphor), it is considered both long and short (i.e., it has representations with both $\sigma = 0$ and $\sigma = 1$).

Lemma 7: Let T be a tree with $f_T \leq 2$. For $\sigma \in \{0, 1\}$ and $M = m - \sigma$, define

$$\underline{c}_\sigma = (N - 2^M)\sigma \quad \text{and} \quad \bar{c}_\sigma = \left\lfloor \frac{2N - 2^M}{3} \right\rfloor . \tag{11}$$

Then, T is equivalent to one of the trees $T_{\sigma,c}$ defined by the profiles

$$\mathbf{N}_{T_{\sigma,c}} = (n_{M-1}, n_M, n_{M+1}) = \Big(2^M - N + c,\ 2N - 2^M - 3c,\ 2c\Big), \quad \sigma \in \{0, 1\}, \ \ \underline{c}_\sigma \leq c \leq \bar{c}_\sigma \,. \tag{12}$$

Remarks.

1) Equation (12) characterizes all trees with N leaves and $f_T \leq 2$ in terms of the parameters σ and c. The parameter c has different ranges depending on σ: we have $N - 2^{m-1} \leq c \leq \lfloor \frac{2N - 2^{m-1}}{3} \rfloor$ when $\sigma = 1$, and $0 \leq c \leq \lfloor \frac{2N-2^m}{3} \rfloor$ when $\sigma = 0$. The use of the parametrized quantities $M, \underline{c}_\sigma$, and $\bar{c}_\sigma$ will allow us to treat the two ranges in a unified way in most cases. Also, notice that $T_{1,\underline{c}_1}$ and $T_{0,\underline{c}_0}$ represent the same tree, corresponding, respectively, to interpretations of the quasi-uniform tree Q_N as short or long.
2) The parameter c represents the number of internal (non-leaf) nodes at level M of T. An increase of c by one corresponds to moving a pair of sibling leaves previously rooted in level $M-1$ to a new parent in level M (thereby increasing the number of internal nodes at that level by one). The number of leaves at level M decreases by three, and the numbers of leaves at levels $M-1$ and $M+1$ increase by one and two, respectively.

[5] Notice that not every 4-uniform source admits an optimal tree with $f_T \leq 2$ (although the ones of interest in this section do). For example, an optimal tree for the 4-uniform source with probabilities $\frac{1}{10}(4, 3, 1, 1, 1)$ must have $f_T > 2$.

Consider now a distribution on N symbols, with associated vector of probabilities (or weights) $\mathbf{p} = (\,p_1, p_2, \ldots, p_N\,)$, $p_1 \geq p_2 \geq \cdots \geq p_N$. Let $L_{\sigma,c}$ denote the average code length of $T_{\sigma,c}$ under $\mathbf{p}$ (with shorter codewords naturally assigned to larger weights), and let

$$D_{\sigma,c} = L_{\sigma,c} - L_{\sigma,c-1}, \quad \sigma \in \{0,1\}, \quad \underline{c}_\sigma < c \leq \overline{c}_\sigma\,. \tag{13}$$

It follows from these definitions, and the structure of the profile (12) (see also Remark 2 above), that

$$D_{\sigma,c} = p_{N-2c+1} + p_{N-2c+2} - p_{2^M-N+c}\,, \quad \sigma \in \{0,1\}, \quad \underline{c}_\sigma < c \leq \overline{c}_\sigma, \tag{14}$$

A useful interpretation of (14) follows directly from the profile (12): for $T_{\sigma,c}$, $D_{\sigma,c}$ is the difference between the sum of the two heaviest weights on level $M+1$ and the lightest weight on level $M-1$.

Let $\mathrm{sg}(x)$ be defined as $-1, 0$, or 1, respectively, for negative, zero, or positive values of x, and consider the following sequence (recalling that $\underline{c}_0 = 0$):

$$\mathbf{s} = -\mathrm{sg}(D_{1,\overline{c}_1}), -\mathrm{sg}(D_{1,\overline{c}_1-1}), \ldots, -\mathrm{sg}(D_{1,\underline{c}_1+1}), \mathrm{sg}(D_{0,1}), \mathrm{sg}(D_{0,2}), \ldots, \mathrm{sg}(D_{0,\overline{c}_0})\,. \tag{15}$$

Lemma 8: The sequence $\mathbf{s}$ is non-decreasing.

The definition of the sequence $\mathbf{s}$ induces a total ordering of the pairs (σ, c) (and, hence, also of the trees $T_{\sigma,c}$), with pairs with $\sigma = 1$ ordered by decreasing value of c, followed by pairs with $\sigma = 0$ in increasing order of c. The two subsequences "meet" at $\underline{c}_\sigma$, which defines the same tree regardless of the value of σ (in the pairs ordering, we take $(1, \underline{c}_1)$ as identical to $(0, \underline{c}_0) = (0,0)$). We denote this total order by $\preceq$. Recalling that the quantities $D_{\sigma,c}$ are differences in average code length between consecutive codes in this ordering, Lemma 8 tells us that, as we scan the codes in order, we will generally see the average code length decrease monotonically, reach a minimum, and then (possibly after staying at the minimum for some number of trees) increase monotonically. In the following theorem, we formalize this observation, and identify the trees $T_{\sigma,c}$ that are optimal for $\mathbf{p}$.

Theorem 3: Let $\mathbf{p}$ be a 4-uniform distribution such that $\mathbf{p}$ has an optimal tree T with $f_T \leq 2$. Define pairs (σ_*, c_*) and (σ^*, c^*) as follows:

$$\begin{aligned} (\sigma_*, c_*) &= (1, \overline{c}_1) \quad \text{if } D_{1,\overline{c}_1} \geq 0\,, \\ (\sigma^*, c^*) &= (0, \overline{c}_0) \quad \text{if } D_{0,\overline{c}_0} \leq 0\,; \end{aligned}$$

otherwise, if $D_{1,\overline{c}_1} < 0$, let (σ_-, c_-) be such that $(-1)^{(\sigma_-)}\mathrm{sg}(D_{\sigma_-,c_-})$ is the last negative entry in $\mathbf{s}$, and define

$$(\sigma_*,\, c_*) \;=\; (\sigma_-,\, c_- - \sigma_-)\,;$$

if $D_{0,\overline{c}_0} > 0$, let (σ_+, c_+) be such that $(-1)^{(\sigma_+)}\mathrm{sg}(D_{\sigma_+,c_+})$ is the first positive entry in $\mathbf{s}$, and define

$$(\sigma^*,\, c^*) \;=\; (\sigma_+,\, c_+ - 1 + \sigma_+)\,.$$

Then, all trees $T_{\sigma,c}$ with $(\sigma_*, c_*) \preceq (\sigma, c) \preceq (\sigma^*, c^*)$ are optimal for $\mathbf{p}$.

TABLE I
EXAMPLE: FINDING OPTIMAL TREES $T_{\sigma,c}$ FOR $N = 19$, $\mathbf{p} = \frac{1}{49}(4,4,3,3,3,3,3,3,3,3,3,2,2,2,2,2,2,1,1)$.
OPTIMAL TREES ARE EMPHASIZED IN BOLDFACE.

(σ, c)	$(1,7)$	$(1,6)$	$(1,5)$	$(\mathbf{1},\mathbf{4})$	$(\mathbf{1},\mathbf{3}) =$ $(\mathbf{0},\mathbf{0})$	$(\mathbf{0},\mathbf{1})$	$(0,2)$
(n_{M-1}, n_M, n_{M+1})	$(4,1,14)$	$(3,4,12)$	$(2,7,10)$	$(\mathbf{1},\mathbf{10},\mathbf{8})$	$(\mathbf{13},\mathbf{6},\mathbf{0})$	$(\mathbf{14},\mathbf{3},\mathbf{2})$	$(15,0,4)$
$49 \cdot L_{\sigma,c}$	214	211	208	**206**	**206**	**206**	208
$49 \cdot D_{\sigma,c}$	3	3	2	0		0	2
$\mathbf{s}$	-1	-1	-1 (σ_-, c_-)	0 (σ_*, c_*)		0 (σ^*, c^*)	1 (σ_+, c_+)

Notice that, by Lemma 8, the range $(\sigma_*, c_*) \preceq (\sigma, c) \preceq (\sigma^*, c^*)$ is well defined and never empty, consistently with the assumptions of the theorem and with Lemma 7. The example in Table I lists all the trees $T_{\sigma,c}$ with $f_T \leq 2$ for $N = 19$, as characterized in Lemma 7, and shows how Theorem 3 is used to find optimal trees for a given 4-uniform distribution on 19 symbols.

C. The top code

By Lemma 5, Theorem 3 applies to the source $\hat{\mathcal{A}}_k$ defined in (10). We will apply the theorem to identify parameters (σ_k, c_k) that yield an optimal tree T_{σ_k,c_k} for $\hat{\mathcal{A}}_k$.

For the remainder of the section, we take $N = k^2$, and let $\mathbf{p} = (p_1, p_2, \ldots, p_{k^2})$ denote the vector of symbol weights in $\hat{\mathcal{A}}_k$, in non-increasing order. For simplicity, we assume that $\mathbf{p}$ is unnormalized, i.e., $\mathbf{p} = (q^0, q^1, q^1, \ldots, q^j, q^j, \ldots, q^j, \ldots, q^{2k-3}, q^{2k-3}, q^{2k-2})$. Here, q^j is repeated $j+1$ times for $0 \leq j \leq k-1$, and $2k-1-j$ times for $k \leq j \leq 2k-2$. The following lemma, which follows immediately from this structure, establishes the relation between indices and weights in $\mathbf{p}$.

Lemma 9: For $0 \leq i < k(k+1)/2$, we have $p_{i+1} = q^j$, where j is the unique integer in the range $0 \leq j \leq k-1$ satisfying

$$i = \frac{j(j+1)}{2} + r \qquad \text{for some } r, \quad 0 \leq r \leq j\,. \tag{16}$$

For $0 \leq i' < k(k+1)/2$, we have $p_{k^2-i'} = q^{2k-2-j'} = \frac{1}{2}q^{k-2-j'}$, where j' is the unique integer in the range $0 \leq j' \leq k-1$ satisfying

$$i' = \frac{j'(j'+1)}{2} + r' \qquad \text{for some } r', \quad 0 \leq r' \leq j'\,. \tag{17}$$

We define some auxiliary quantities that will be useful in the sequel. Let $m = \lceil \log k^2 \rceil$, $Q = k^2 - \lceil k(k-1)/4 \rceil$, and $M' = \lceil \log_2 Q \rceil$, with dependence on k understood from the context. We assume that $k > 2$, since the optimal codes for $k = 1$ and $k = 2$ have already been described in Subsection IV-A. It is readily verified that we must have either $M' = m$ or $M' = m-1$. The next lemma shows that the relation between M' and m determines the parameter σ of the optimal trees $T_{\sigma,c}$ for $\hat{\mathcal{A}}_k$.

Lemma 10: If $M' = m$, then trees $T_{\sigma,c}$ that are optimal for $\hat{\mathcal{A}}_k$ are long ($\sigma = 0$); otherwise, they are short ($\sigma = 1$).

Proof: Assume $M' = m$. Then, we can write

$$2^m = 2^{M'} < 2^{1+\log Q} = 2Q = 2k^2 - 2\lceil k(k-1)/4 \rceil \leq 2k^2 - k(k-1)/2\,, \tag{18}$$

so $2^m - k^2 < k^2 - k(k-1)/2$. If $\underline{c}_1 + 1 > \overline{c}_1$, then all trees $T_{\sigma,c}$ in (12) are long. Otherwise, $D_{1,\underline{c}_1+1}$ is well defined, and we have

$$\begin{aligned} -D_{1,\underline{c}_1+1} &= -D_{1,k^2-2^{m-1}+1} = p_1 - (p_{2^m-k^2-1} + p_{2^m-2^k}) \\ &\leq p_1 - 2p_{k^2-k(k-1)/2} = p_1 - 2q^{k-1} = 1 - q^{-1} < 0\,, \end{aligned} \tag{19}$$

where the first and second equalities follow from the definition of $\underline{c}_1$ and from (14), the first inequality from the ordering of the weights and from (18), the third equality from Lemma 9, and the last equality from the relation $q^k = \frac{1}{2}$. By Lemma 8, we conclude that optimal trees for $\hat{\mathcal{A}}_k$ are long in this case. Similarly, when $M' = m - 1$, we have

$$2^m \geq 2Q \geq 2k^2 - k(k-1)/2 - 2\,, \tag{20}$$

so $2^m - k^2 + 1 \geq k^2 - k(k-1)/2 - 1$, and $p_{2^m-k^2+1} \leq p_{k^2-k(k-1)/2-1} = q^k = \frac{1}{2}$. If $\overline{c}_0 = \underline{c}_0 = 0$, then all trees $T_{\sigma,c}$ in (12) are short. Otherwise, similarly to (19), we have

$$D_{0,1} = p_{k^2-1} + p_{k^2} - p_{2^m-k^2+1} > 2q^{2k-2} - \frac{1}{2} = \frac{q^{-2}}{2} - \frac{1}{2} > 0,$$

which implies that optimal trees are short in this case. ■

It follows from Lemma 10 that we can take $m - M'$ as the parameter σ for all trees $T_{\sigma,c}$ that are optimal for $\mathbf{p}$. Notice that M' is analogous to the parameter M defined in Lemma 7, but slightly stricter, in that, in cases where a quasi-uniform tree is optimal, $m - M'$ will assume a definite value in $\{0, 1\}$ (which will vary with k), while, in principle, a representation with either value of σ is available. This very slight loss of generality is of no consequence to our derivations, and, in the sequel, we will identify M with M', i.e., we will take $M = \lceil \log Q \rceil$. It also follows from Lemma 10 that when applying Theorem 3 to find optimal trees for $\mathbf{p}$, we only need to focus on one of the two segments (corresponding to σ=0 or σ=1) that comprise the sequence $\mathbf{s}$ in (15), the choice being determined by the value of k. This will simplify the application of the theorem.

Lemmas 9 and 10, together with Theorem 3, suggest a clear way, at least in principle, for finding an optimal tree $T_{\sigma,c}$ for $\hat{\mathcal{A}}_k$. The parameter σ is determined immediately as $\sigma = m - M$ (recalling that m and M are determined by k). Now, recalling the expression for $D_{\sigma,c}$ in (14), we observe that as c increases, the weights p_{k^2-2c+1} and p_{k^2-2c+2} also increase, while $p_{2^M-k^2+c}$, which gets subtracted, decreases. Thus, since, by Theorem 3, an optimal value of c occurs when $D_{\sigma,c}$ changes sign, we need to search for the value of c for which the increasing sum of the first two terms "crosses" the value of the decreasing third term. This can be done, at least roughly, by using explicit weight values from Lemma 9 with $i' \in \{2c-1,\, 2c-2\}$ and $i = 2^m - k^2 + c$, and solving a quadratic equation, say, for the parameter j (the parameter j' will be tied to j by the constraint $D_{\sigma,c} \approx 0$). A finer adjustment of the solution

TABLE II
EXAMPLES OF OPTIMAL CODE PARAMETERS AND PROFILES FOR $\hat{\mathcal{A}}_k,\ 3 \le k \le 10$.

k	M	j	r	σ_k	c_k	(n_{M-1}, n_M, n_{M+1})
2	2	0	0	0	0	$(0, 4, 0)$
3	3	0	0	1	1	$(0, 7, 2)$
4	4	1	0	0	1	$(1, 13, 2)$
5	5	3	1	0	0	$(7, 18, 0)$
6	5	1	0	1	5	$(1, 25, 10)$
7	6	5	0	0	0	$(15, 34, 0)$
8	6	2	2	0	5	$(5, 49, 10)$
9	6	0	0	1	17	$(0, 47, 34)$
10	7	7	1	0	1	$(29, 69, 2)$

is achieved with the parameters r and r', observing that a change of sign of $D_{\sigma,c}$ can only occur near locations where the weights in $\mathbf{p}$ change (i.e., "jumps" in either j or j'), which occur at intervals of length up to k. At the "jump" locations, either r or r' must be close to zero. While there is no conceptual difficulty in these steps, the actual computations are somewhat involved, due to various integer constraints and border cases. Theorem 4 below takes these complexities into account and characterizes, explicitly in terms of k, the parameter pair (σ_k, c_k) of an optimal code T_{σ_k, c_k} for $\hat{\mathcal{A}}_k$.

Theorem 4: Let $q = 2^{-1/k}$, $Q = k^2 - \lceil k(k-1)/4 \rceil$, $m = \lceil \log k^2 \rceil$, and $M = \lceil \log Q \rceil$. Define the function

$$\Delta(x) = 2k^2 - 2^{M+1} + x(x+1) - \frac{(k-x-2)(k-x-1)}{2}\,. \tag{21}$$

Let x_0 denote the largest real root of $\Delta(x)$, and let $\xi = \lfloor x_0 \rfloor$. Set

$$j = \xi, \qquad r = \left\lfloor \frac{-\Delta(j)+1}{2} \right\rfloor \quad \text{when} \quad -\Delta(\xi) \le 2\xi, \tag{22}$$

$$j = \xi + 1, \quad r = 0 \qquad \text{otherwise.} \tag{23}$$

Then, the tree T_{σ_k,c_k}, as defined by the profile (12) with $\sigma = \sigma_k = m - M$ and

$$c = c_k = k^2 - 2^M + \frac{j(j+1)}{2} + r\,, \tag{24}$$

is optimal for $\hat{\mathcal{A}}_k$. Furthermore, c_k is the smallest value of c for any optimal tree $T_{\sigma_k,c}$ for $\hat{\mathcal{A}}_k$.

The proof of Theorem 4 is presented in Appendix B. In the theorem (and its proof), we have chosen to identify the optimal tree $T_{\sigma_k,c}$ with the *smallest* possible value of c. It can readily be verified that this choice minimizes the *variance* of the code length among all optimal trees $T_{\sigma_k,c}$. With only minor changes in the construction and proof, one could also identify the *largest* value of c for an optimal tree, and, thus, the full range of values of c yielding optimal trees $T_{\sigma_k,c}$. For conciseness, we have omitted this extension of the proof.

Examples of the application of Theorem 4 are presented in Table II, which lists the parameters M, j, r, σ_k, c_k, and the profile of the optimal tree T_{σ_k, c_k} defined by the theorem, for $3 \le k \le 10$.

The tools derived in the proof of Theorem 4 also yield the following result, a proof of which is also presented in Appendix B.

Corollary 1: Let $k > 2$ and $q = 2^{-1/k}$. Then, $G_k \cdot G_k$ is not optimal for TDGD(q).

D. Average code length

Corollary 2: Let M, $\Delta(x)$, j, and r be as defined in Theorem 4. Then, the average code length $\mathcal{L}_q(C_k)$ for the code C_k under TDGD(q), for arbitrary q, is given by

$$\mathcal{L}_q(C_k) = M + 1 + \\ + \frac{q^j \left(1 - q^{k+1} + (1-q)\left(q^{k+1}\,(k-j-1) + j\right) + (1-q)^2\left(q^k\left(\,2\,r + \Delta(j)\,\right) - r\right)\right)}{(1-q^k)^2}\,. \tag{25}$$

When $q = 2^{\,-1/k}$, we have

$$\mathcal{L}_q(C_k) = M + 1 + 2\,q^j\left(1 + (1-q)\,(q\,k + (2-q)\,j) + (1-q)^2\,(1 + \Delta\,(j))\right) \tag{26}$$

Proof: By Theorem 2, the code length for (a, b) under C_k is $|T_k(a \bmod k, b \bmod k)| + 2 + \lfloor \frac{a}{k} \rfloor + \lfloor \frac{b}{k} \rfloor$. Writing $a = mk + i$ and $b = nk + j$ with $0 \le i, j < k$, $m, n \ge 0$, the average code length under C_k is

$$\begin{aligned} \mathcal{L}_q(C_k) &= \sum_{0 \le i,j < k}\ \sum_{m,n \ge 0} q^{i+j+(m+n)k}\,(\,|T_k(a, b)| + m + n + 2\,) \\ &= \frac{2}{1-q^k} + \frac{(1-q)^2}{(1-q^k)^2} \sum_{0 \le i,j \le k-1} |T_k(i, j)|\,q^{i+j} = \frac{2}{1-q^k} + \mathcal{L}_q(T_k)\,, \end{aligned} \tag{27}$$

where the second equality follows from elementary series computations, and the third identifies the (normalized) average code length of the code T_k defined in Theorem 4. Denote by W_{M-1}, W_M, and W_{M+1} the total normalized weight of symbols in $\hat{\mathcal{A}}_k$ assigned length $M-1$, M, and $M+1$, respectively, by T_k. Then, the average code length of T_k is given by

$$\mathcal{L}_q(T_k) = (M-1)\,W_{M-1} + M\,W_M + (M+1)W_{M+1} = M + W_{M+1} - W_{M-1}\,. \tag{28}$$

From the profile (12), with $N = k^2$ and $c = c_k$ as defined in (24), recalling (16), letting $\gamma = (1-q)^2/(1-q^k)^2$, and carrying out the computations, we obtain

$$W_{M-1} = \gamma \sum_{i=1}^{j(j+1)/2+r} p_i = \gamma \sum_{\ell=0}^{j-1} (\ell+1) q^\ell + \gamma\,r\,q^j = \frac{1 - q^j\left(1 + (1-q)j - (1-q)^2 r\right)}{(1-q^k)^2}\,.$$

Similarly, from the proof of Theorem 4, setting $j' = k - j - 2$ and $r' = 2r + \Delta(j)$, we obtain

$$\begin{aligned} W_{M+1} &= \gamma \sum_{i=0}^{j'(j'+1)/2+r'-1} p_{k^2-i} = \gamma \sum_{\ell=0}^{j'-1} (\ell+1) q^{2k-2-\ell} + \gamma\,r'\,q^{2k-2-j'} \\ &= \frac{q^{2\,k} + q^{k+j+1}\,((k-j-1)(1-q) - 1) + q^{k+j}(1-q)^2\,(2\,r + \Delta(j))}{(1-q^k)^2}\,. \end{aligned}$$

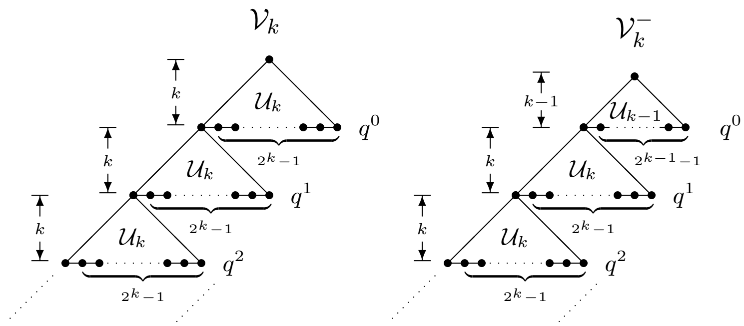


Fig. 4. Trees $\mathcal{V}_k$ and $\mathcal{V}_k^-$.

The result (25) now follows by substituting the above expressions for W_{M-1} and W_{M+1} in (28), substituting for $\mathcal{L}_q(T_k)$ in (27), and using appropriate algebraic simplifications. The result (26), in turn, follows by applying the relation $q^k = 1/2$. ■

V. Optimal codes for TDGDs of parameter $q = 2^{-k}$

A. The codes

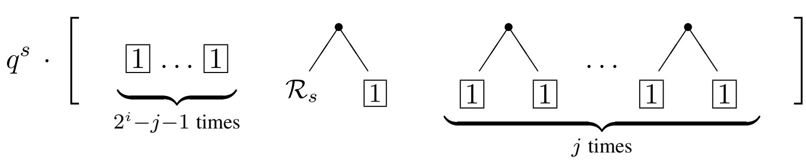
Assume $q = 2^{-k}$ for some integer $k > 1$. We reuse the notation $\mathcal{U}_m = Q_{2^m}$ for a uniform tree of depth m, assuming, additionally, that its 2^m leaves have weight one. The infinite tree (and associated multiset of leaf weights) $\mathcal{V}_k$ is recursively defined as follows. Start from $\mathcal{U}_k$, and attach to its leftmost leaf a copy of $q\mathcal{V}_k$. Thus, $\mathcal{V}_k$ has 2^k-1 leaves of weight q^s at depth $(s+1)k$ for all $s \geq 0$, and no other leaves. The related tree $\mathcal{V}_k^-$ is defined by starting from $\mathcal{U}_{k-1}$, and attaching to its leftmost leaf a copy of $q\mathcal{V}_k$. Thus, $\mathcal{V}_k^-$ has $2^{k-1}-1$ leaves of weight q^0 at depth $k-1$, and 2^k-1 leaves of weight q^s at depth $(s+1)k-1$ for all $s > 0$. The trees $\mathcal{V}_k$ and $\mathcal{V}_k^-$ are illustrated in Figure 4.

We describe a sequence of binary trees (and codes) C_{-k}, which, later in the section, will be shown to be optimal for TDGDs with $q = 2^{-k}$, $k > 1$. We describe the trees by *layers*. A layer $\mathbf{L}_s$ is a collection of consecutive levels of the tree, containing all the leaves with signature s. The structure of the layers, and how $\mathbf{L}_s$ unfolds into $\mathbf{L}_{s+1}$ for all s, are presented next, providing a full description of the trees C_{-k}.

Assume $k > 1$ is fixed. We distinguish two main cases for the structure of $\mathbf{L}_s$, which depend on the value of s, as specified below. In the description of the layers, each tree structure is a virtual symbol. We will refer to both original and virtual symbols simply as *symbols*.

Case 1) $0 \leq s \leq 2^{k-1} - 2$:

Write $s = 2^i + j - 1$ with $0 \leq i \leq k-2$, $0 \leq j \leq 2^i - 1$. Layer $\mathbf{L}_s$ consists of nodes in two levels, arranged as follows:



(29)

The symbol $\mathcal{R}_s$ represents a tree containing all the signatures strictly greater than s, scaled by q^{-s}. Layer $\mathbf{L}_s$ emerges from constructing a quasi-uniform tree for $s+2$ symbols ($s+1$ signatures s, and the symbol $\mathcal{R}_s$), attached to $\mathcal{R}_{s-1}$ of the previous layer if $s > 0$, or to the root of the tree if $s = 0$. We have $s+2 = 2^i + 1 + j$, $0 \le j \le 2^i - 1$, so the quasi-uniform tree has $2^i - j - 1$ leaves at depth i, and $2j+2$ leaves at level $i+1$, as shown in (29).

Case 2) $s \ge 2^{k-1} - 1$:

Write

$$s = 2^{k-1} - 1 + (2^k - 1)\ell + j, \qquad \text{with } \ \ell \ge 0, \quad 0 \le j < 2^k - 1\,. \tag{30}$$

There are five types of layers in this case, as described below. The symbol $\mathcal{R}_s$ in each case represents a tree containing all the signatures strictly greater than s that are not contained in other virtual symbols in $\mathbf{L}_s$, suitably scaled by q^{-s}. Also, it will be convenient to use the notation $\boldsymbol{\mathcal{M}}$ as shorthand for the sequence

$$\boldsymbol{\mathcal{M}}: \quad q\mathcal{V}_k\,,\ \underbrace{\boxed{1}\dots\boxed{1}}_{2^k-1 \text{ times}} \tag{31}$$

(nevertheless, $\boldsymbol{\mathcal{M}}$ counts as 2^k symbols in $\mathbf{L}_s$).

(i) $\quad 0 \le j \le 2^{k-1}{-}3$ (for $k > 2$):

$$q^s \cdot \Big[\ \underbrace{\boldsymbol{\mathcal{M}}\ \dots\ \boldsymbol{\mathcal{M}}}_{\ell \text{ times}}\ \underbrace{\boxed{1}\dots\boxed{1}}_{2^{k-1}-j-1 \text{ times}}\ \overset{\bullet}{\underset{\mathcal{R}_s\quad\boxed{1}}{\diagup\diagdown}}\ \underbrace{\overset{\bullet}{\underset{\boxed{1}\quad\boxed{1}}{\diagup\diagdown}}\ \dots\ \overset{\bullet}{\underset{\boxed{1}\quad\boxed{1}}{\diagup\diagdown}}}_{j \text{ times}}\ \Big] \tag{32}$$

(ii) $\quad j = 2^{k-1}{-}2$:

$$q^s \cdot \Big[\ \underbrace{\boldsymbol{\mathcal{M}}\ \dots\ \boldsymbol{\mathcal{M}}}_{\ell \text{ times}}\ \overset{\bullet}{\underset{q\,\mathcal{U}_{k-1}\quad\mathcal{R}_s}{\diagup\diagdown}}\ \underbrace{\overset{\bullet}{\underset{\boxed{1}\quad\boxed{1}}{\diagup\diagdown}}\ \dots\ \overset{\bullet}{\underset{\boxed{1}\quad\boxed{1}}{\diagup\diagdown}}}_{2^{k-1}-1 \text{ times}}\ \Big] \tag{33}$$

(iii) $\quad 2^{k-1} - 1 \le j \le 2^k{-}4$:

$$q^s \cdot \Big[\ \underbrace{\boldsymbol{\mathcal{M}}\ \dots\ \boldsymbol{\mathcal{M}}}_{\ell \text{ times}}\ \underbrace{\boxed{1}\dots\boxed{1}}_{3\cdot 2^{k-1}-2-j \text{ times}}\ \overset{\bullet}{\underset{q\,\mathcal{U}_{k-1}\quad\mathcal{R}_s}{\diagup\diagdown}}\ \underbrace{\overset{\bullet}{\underset{\boxed{1}\quad\boxed{1}}{\diagup\diagdown}}\ \dots\ \overset{\bullet}{\underset{\boxed{1}\quad\boxed{1}}{\diagup\diagdown}}}_{j-2^{k-1}+1 \text{ times}}\ \Big] \tag{34}$$

(iv) $\quad j = 2^k{-}3$:

$$q^s \cdot \Big[\ \underbrace{\boldsymbol{\mathcal{M}}\ \dots\ \boldsymbol{\mathcal{M}}}_{\ell \text{ times}}\ \underbrace{\boxed{1}\dots\boxed{1}}_{2^{k-1}+1 \text{ times}}\ \overset{\bullet}{\underset{q\mathcal{V}_k^-\quad\mathcal{R}_s}{\diagup\diagdown}}\ \underbrace{\overset{\bullet}{\underset{\boxed{1}\quad\boxed{1}}{\diagup\diagdown}}\ \dots\ \overset{\bullet}{\underset{\boxed{1}\quad\boxed{1}}{\diagup\diagdown}}}_{2^{k-1}-2 \text{ times}}\ \Big] \tag{35}$$

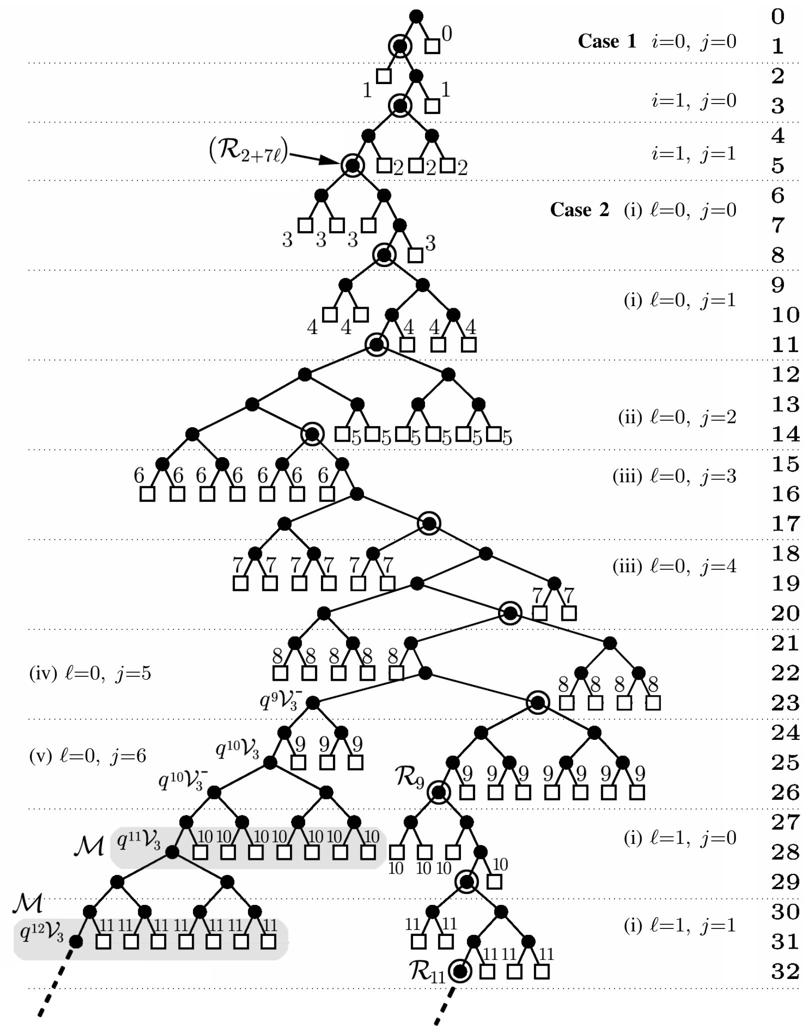


Fig. 5. Top layers of the optimal tree C_{-3} ($q = \frac{1}{8}$), with leaf signatures noted for $s \leq 11$. Dotted lines separate layers $\mathbf{L}_s$, and circled nodes represent roots of trees $\mathcal{R}_s$.

(v) $j = 2^k - 2$:

$$q^s \cdot \left[\ \underbrace{\mathcal{M}\ \dots\ \mathcal{M}}_{\ell\text{ times}} \quad q\mathcal{V}_k \quad \underbrace{\boxed{1}\dots\boxed{1}}_{2^{k-1}-1\text{ times}} \quad \mathcal{R}_s \ \ \boxed{1} \quad \underbrace{\boxed{1}\ \ \boxed{1}\ \ \dots\ \ \boxed{1}\ \ \boxed{1}}_{2^{k-1}-1\text{ times}}\ \right] \tag{36}$$

The last layer from Case 1 contains all the signatures $s' = 2^{k-1} - 2$. All signatures $s > s'$ are contained in $\mathcal{R}_{s'}$. In particular, there are 2^{k-1} signatures $s' + 1 = 2^{k-1} - 1$. Assume $k > 2$. A quasi-uniform tree with $2^{k-1} + 1$ leaves is constructed, rooted at $\mathcal{R}_{s'}$. This tree has $2^{k-1} - 1$ leaves labeled $s' + 1$ at depth $k - 1$ from its root, and two leaves at depth k, one of which is labeled $s' + 1$, and one that serves as the root for $\mathcal{R}_{s'+1}$. This is consistent with the structure of the first layer in Case 2 shown in (32), with $s = s' + 1$, $\ell = 0$ and $j = 0$. From that layer on, layers of types (i)–(v) above unfold in a cyclic way,

TABLE III

CODE LENGTHS AND CODEWORD COUNTS FOR CODES C_{-k} ON SIGNATURES s, $0 \le s \le 2^{k-1}-2$.

Case 1: $0 \le s \le 2^{k-1}-2, \quad s = 2^i + j - 1, \quad 0 \le i \le k-2$ $\Lambda_s = (s+2)(i+1) - 2^{i+1}$		
	Number of codewords (signatures)	
Range of j	length Λ_s	length Λ_s+1
$0 \le j \le 2^i - 1$	$(2^i - j - 1)$	$2j+1$

TABLE IV

CODE LENGTHS AND CODEWORD COUNTS FOR CODES C_{-k} ON SIGNATURES $s \ge 2^{k-1}-1$.

Case 2: $s \ge 2^{k-1}-1, \quad s = 2^{k-1} - 1 + (2^k-1)\ell + j, \quad \ell \ge 0$ $\Lambda_s = (s+2)k - 2^k$		
	Number of codewords (signatures)	
Range of j	length Λ_s	length Λ_s+1
$0 \le j \le 2^{k-1}-3$	$(2^k-1)\ell + (2^{k-1} - j - 1)$	$2^j + 1$
$j = 2^{k-1} - 2$	$(2^k-1)\ell$	$2^k - 2$
$2^{k-1} - 1 \le j \le 2^k - 4$	$(2^k-1)\ell + 3{\cdot}2^{k-1} - 2 - j$	$2j + 2 - 2^k$
$j = 2^k - 3$	$(2^k-1)\ell + 2^{k-1} + 1$	$2^k - 4$
$j = 2^k - 2$	$(2^k-1)\ell + 2^{k-1} - 1$	$2^k - 1$

each cycle corresponding to an increment in the value of ℓ.

When $k = 2$, layers of type (i) or (iii) are not used. In this case, the only layer in Case 1 contains the signature 0. A uniform tree $\mathcal{U}_2$ is constructed, rooted at $\mathcal{R}_0$. One pair of sibling leaves is assigned to signature 1, while the other pair is assigned to $\mathcal{R}_1$ and $\mathcal{U}_1$, attaining a configuration of type (ii) in Case 2. From that point on, the cyclic layer sequence is (ii)→(iv)→(v)→(ii).

The fine details of the various layer transitions are given in Appendix C, and are illustrated in Figure 5, which shows the upper part of the tree for $k = 3$.

The cyclic nature of the construction is reflected in the structure of the tree. Thus, the subtree $\mathcal{R}_s$, $s \ge 2^{k-1} - 2$ is identical to all subtrees $\mathcal{R}_{s+(2^k-1)\ell'}$, $\ell' \ge 0$, up to appropriate scaling by $q^{(2^k-1)\ell'}$. In the example of Figure 5, the tree $\mathcal{R}_9$ is identical to the tree $\mathcal{R}_2$, indicated in the figure as $\mathcal{R}_{2+7\ell}$. An additional source of self-similarity is provided by the trees $\mathcal{V}_k$ and $\mathcal{V}_k^-$; in Figure 5, the sub-tree labeled $q^{10}\mathcal{V}_3^-$ is identical to that labeled $q^9\mathcal{V}_3^-$, etc. Overall, although the width of the tree is unbounded (driven by the ℓ copies of $\boldsymbol{\mathcal{M}}$ in each layer of Case 2), the total number of distinct sub-trees in C_{-k} is finite.

The following theorem enumerates the code lengths assigned to signatures by the codes C_{-k}. It follows immediately from the description of the codes in (29) and (32)–(36).

Theorem 5: Code C_{-k}, $k{>}1$, assigns code lengths Λ_s or $\Lambda_s + 1$ to signatures s according to the expressions for Λ_s and the codeword counts in Tables III and IV, corresponding, respectively, to the cases $0 \le s \le 2^{k-1} - 2$ (Case 1) and $s \ge 2^{k-1} - 1$ (Case 2).

We now present some auxiliary results that will be useful in proving the optimality of the codes C_{-k}.

We rely on the following relations, which are readily derived from the definitions of the respective trees, under the assumption $q = 2^{-k}$.

$$w(\mathcal{U}_k) = 2w(\mathcal{U}_{k-1}) = w(\mathcal{V}_k) = 2w(\mathcal{V}_k^-) = q^{-1}\,. \tag{37}$$

The next lemma bounds the weight of the symbol $\mathcal{R}_s$ in (29) and (32)–(36).

Lemma 11: When $s \le 2^{k-1} - 2$ (Case 1), we have $0 \le w(\mathcal{R}_s) \le \frac{7}{9}$. When $s > 2^{k-1} - 2$ (Case 2), we have $\frac{1}{2} \le w(\mathcal{R}_s) \le 1$.

Proof: For $s \le 2^{k-1} - 2$, we have

$$w(\mathcal{R}_s) = \sum_{s'=s+1}^{\infty} (s'+1)q^{-s}w(s') = \sum_{r=0}^{\infty}(s+r+2)q^{r+1} = \frac{(s+1)(1-q)+1}{(1-q)^2}\,q\,. \tag{38}$$

The right-hand side of (38) increases with s. Setting $s = 2^{k-1} - 2 = \frac{1}{2q} - 2$, we obtain $w(\mathcal{R}_s) = \frac{1}{2}\left(1 + \frac{q(1+q)}{(1-q)^2}\right)$, which satisfies the claimed upper bound for $q \le \frac{1}{4}$. When $s \ge 2^{k-1} - 1$, $\mathcal{R}_s$ contains all the signatures $s' > s$ (with their weights scaled by q^{-s}) that are not contained in the components $q\mathcal{V}_k$ of the groups $\boldsymbol{\mathcal{M}}$, or in a possible sibling $q\mathcal{U}_{k-1}$ or $q\mathcal{V}_k^-$ of $\mathcal{R}_s$. Write s as in (30). The scaled total weight of signatures $s' > s$ is

$$W_s = q^{-s}\sum_{r=0}^{\infty}(s+2+r)q^{s+1+r} = \frac{(s+2)q}{1-q} + \frac{q^2}{(1-q)^2} = \frac{2q(1+j)+1}{2(1-q)} + \frac{q^2}{(1-q)^2} + \ell\,, \tag{39}$$

where the last equality follows by applying (30) and substituting q^{-1} for 2^k. Let W_s' denote the part of W_s that is contained in the symbols $q\mathcal{V}_k$, $q\mathcal{U}_{k-1}$, or $q\mathcal{V}_k^-$ mentioned above. Observing the layer structures in (32)–(36), and applying (37), we obtain $W_s' = \ell + \delta$, where:

$$\delta = \begin{cases} 0, & 0 \le j \le 2^{k-1} - 3, \\ \frac{1}{2}, & 2^{k-1} - 2 \le j \le 2^k - 3, \\ 1, & j = 2^k - 2\,. \end{cases} \tag{40}$$

The claim of the lemma for $s > 2^{k-1} - 2$ follows by writing $w(\mathcal{R}_s) = W_s - W_s'$, observing that $w(\mathcal{R}_s)$ increases monotonically with j, and bounding $w(\mathcal{R}_s)$, as an elementary function of q , in the interval $0 < q \le \frac{1}{4}$ for each of the cases in (40). Notice that due to the mentioned monotonicity, $w(\mathcal{R}_s)$ is evaluated only at the ends of the ranges of j in (40), and we substitute q^{-1} for 2^k. ∎

The following is an immediate consequence of Lemma 11.

Corollary 3: Let $\mathcal{R}_s'$ denote the virtual symbol containing $\mathcal{R}_s$ in each layer $\mathbf{L}_s$ listed in (29) and (32)–(36). Then, after scaling by q^{-s}, all the symbols to the left of $\mathcal{R}_s'$ in $\mathbf{L}_s$ are of weight 1, all the symbols to its right are of weight 2, and we have $1 \le w(\mathcal{R}_s') \le 2$.

Proof: The claims on the symbols to the left and to the right of $\mathcal{R}_s'$ follow from (37) and the definition of the notation $\boldsymbol{\mathcal{M}}$ in (31). As for $\mathcal{R}_s'$, we have $w(\mathcal{R}_s') = 1 + w(\mathcal{R}_s)$, and the claim of the corollary follows by applying Lemma 11. ∎

Theorem 6: The prefix code C_{-k} is optimal for TDGD(q) with $q = 2^{-k}$, $k > 1$.

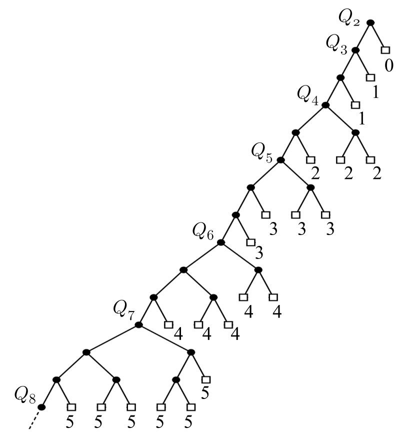


Fig. 6. Top of the limit tree $C_{-\infty}$.

Proof: As before, we rely on the method from [2]. The reduced sources are defined by the layers $\mathbf{L}_s$ defined in (29) and (32)-(36). The steps taking a reduced source to one of lower order follow the "unfolding" steps listed in the description of the codes C_{-k}, in reverse order (bottom-up). It remains to show that these steps correspond to a valid sequence of mergers in the Huffman procedure. Consider a layer $\mathbf{L}_s$, and let $\psi_1, \psi_2, \ldots, \psi_N$ denote its symbols, listed from left to right. It is readily verified, by observing (29) and (32)-(36), that $N = 2^i$ for a layer (29), with i as defined in Case 1, and that N is divisible by 2^{k-1} in layers of type (i)–(ii), and by 2^k in layers of type (iii)–(v). By Corollary 3, the ψ_j are ordered by increasing weight order, and the merger of any two of them results in a combined weight that is at least as large as any weight in the layer. Thus, merging ψ_{2j-1} with ψ_{2j}, $1 \leq j \leq N/2$, is a valid sequence of steps in the Huffman procedure on $\mathbf{L}_s$. Moreover, since there is at most one symbol of weight different from 1 or 2 (after scaling), and strictly between them, the resulting sequence of merged weights includes weights 2, ω, and 4, with $2 \leq \omega \leq 4$, with at most one symbol of weight ω. We iterate the argument, until we reach layer $\mathbf{L}_{s-1}$. ■

B. A limit code

The sequence of optimal codes C_{-k} stabilizes in the limit of $k \to \infty$ ($q \to 0$), as stated in the following corollary.

Corollary 4: When $k{\to}\infty$, the sequence of optimal trees C_{-k} converges to a limit tree $C_{-\infty}$ that can be constructed as follows: start with Q_n for $n{=}2$, recursively replace the leftmost leaf of the deepest level of the current tree by Q_{n+1}, and increase n.

Proof: The corollary is proved by observing that the part of the tree corresponding to $0 \leq s \leq 2^{k-1}$ in Theorem 6 remains invariant for all $k' \geq k$. This corresponds to the layers $\mathbf{L}_s$ of Case 1. ■

The limiting property of $C_{-\infty}$ in connection with the TDGD is mentioned also in [10, Ch. 5]. Figure 6 shows the first fourteen levels of $C_{-\infty}$. Notice that the first eleven levels coincide with those of C_{-3} in Figure 5, up to reordering of nodes at each level. Explicit encoding with $C_{-\infty}$ can be done as follows. Given a pair (i, j), with signature $s = i{+}j$, we write $s = 2^t - 1 + r$, with $0 \leq r \leq 2^t - 1$ and $t \geq 0$. We encode (i,j) with a binary codeword xy, where $x = 1^{(t-1)(s+1)+2r+1}$ identifies the path to the root of the quasi-uniform tree that contains all the leaves of signature s, and $y = Q_{s+2}(i+1)$. The resulting code length distribution for signature s is: $2^t - 1 - r$ signatures encoded with length $(t-1)(s+2)+2r+2$, $2r+1$ signatures encoded with length $(t-1)(s+2)+2r+3$.

The following corollary shows the average code length attained by $C_{-\infty}$ on an arbitrary TDGD.

Corollary 5: The average code length of the limit code $C_{-\infty}$ under TDGD(q) is given by

$$\mathcal{L}_q(C_{-\infty}, q) = 1 + \frac{1}{1-q}\sum_{t\geq 0} q^{2^t}\left(2^t(1-q)+2\right).$$

Proof: For $s \geq 0$, let r and t, $t \geq 0$, $0 \leq r \leq 2^t - 1$, be the (uniquely determined) integers such that $s = 2^t - 1 + r$. By Corollary 4 and the ensuing discussion, we can write

$$\begin{aligned}
\mathcal{L}_q(C_{-\infty}, q) &= (1-q)^2 \sum_{t\geq 0}\sum_{s=2^t-1}^{2^{t+1}-2}\Big(\big((t-1)(s+2)+2r+2\big)(s+1)+2r+1\Big)q^s && (41)\\
&= (1-q)^2 \sum_{t\geq 0}\Big(q^{2^{t+1}-1}A(t) + q^{2^t-1}B(t)\Big)\,, && (42)
\end{aligned}$$

for functions $A(t)$ and $B(t)$ resulting from substituting $r = s - 2^t + 1$ and carrying out the inner summation in (41). It can be verified, by symbolic manipulation, that

$$B(0) = \frac{1-q^2+2q}{(1-q)^3}\,, \qquad \text{and} \qquad A(t-1)+B(t) = q\frac{2^t - 2^t q + 2}{(1-q)^3}\,.$$

Substituting in (42), after rearranging terms, we obtain

$$\begin{aligned}
\mathcal{L}_q(C_{-\infty}, q) &= (1-q)^2\left(B(0) + \sum_{t\geq 1} q^{2^t-1}\Big(A(t-1)+B(t)\Big)\right)\\
&= (1-q)^2\left(\frac{1-q^2+2q}{(1-q)^3} + \sum_{t\geq 1} q^{2^t}\frac{2^t - 2^t q + 2}{(1-q)^3}\right)\\
&= 1 + \frac{1}{1-q}\sum_{t\geq 0} q^{2^t}\left(2^t(1-q)+2\right).
\end{aligned}$$

■

VI. Practical considerations and redundancy

In a practical situation, one could use the codes C_k for $q \geq \frac{1}{2}$, and the codes C_{-k} for $q < \frac{1}{2}$. However, a lower complexity alternative, which incurs a modest code length penalty (as shown in Figure 7), is to use $C_{-\infty}$ in lieu of the codes C_{-k}, up to the value of q where switching to C_1 gives better average code length. The crossover point is at $q \approx 0.33715$.

Encoding a symbol pair (x,y) with a code C_k is of about the same complexity as two encodings of individual symbols with a Golomb code of order k. As described in Theorem 2, the encoding with

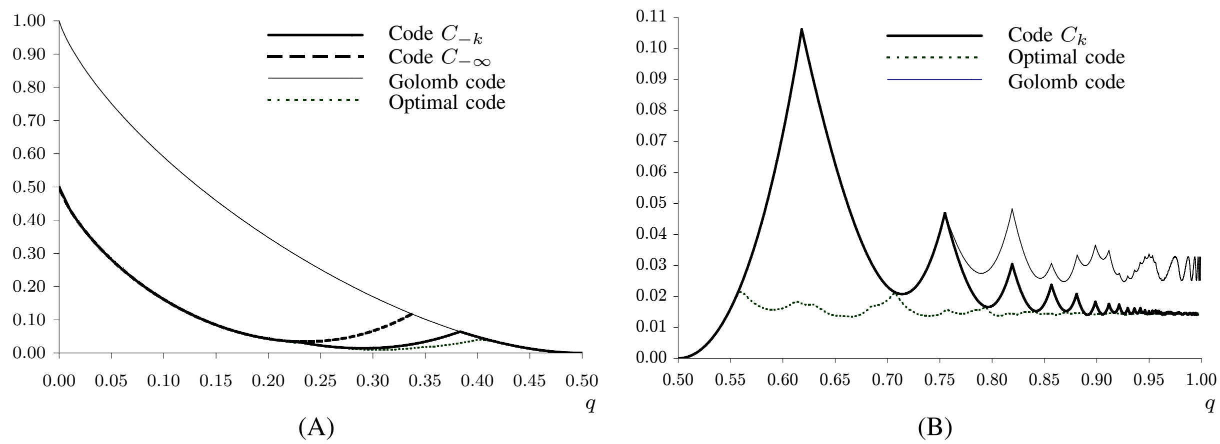


Fig. 7. Redundancy (in bits/integer symbol) for the optimal prefix code (estimated numerically), the best Golomb code, the limit code $C_{-\infty}$, and the best code C_{-k} or C_k for each value of q, (A) $0 < q < \frac{1}{2}$, (B) $\frac{1}{2} \leq q < 1$. The limit code $C_{-\infty}$ is plotted up to $q = 0.33715\ldots$, where its curve intersects that of C_1.

C_k entails unary encodings of $\lfloor x/k \rfloor$ and $\lfloor y/k \rfloor$, which would also be needed with the Golomb code. Given the profile of the top code $T_k = T_{\sigma_k, c_k}$, determined in Theorem 4, encoding with T_k requires comparing the index of the pair $(x \bmod k, y \bmod k)$ with at most two fixed thresholds, to determine the corresponding code length (which can assume up to three consecutive integer values). The codeword is then computed directly from the index. Each encoding with the Golomb code, on the other hand, requires one comparison with a fixed threshold to determine the code length of each Q_k component, or a total of two for the pair (x, y).

As in the one-dimensional case (see, e.g., [3], [13]), when encoding a sequence $x_1, x_2, \ldots, x_{2t}, \ldots$, the best code for the next pair (x_{2t-1}, x_{2t}) can be determined adaptively, driven by the sufficient statistic $S_t = t^{-1} \sum_{j=1}^{2t-2} x_j$. The crossover points for the estimates of the code parameter k can be precomputed and stored in terms of the statistic S_t. The one-dimensional code has a slight advantage in the adaptation, in that it can adapt its statistic with every symbol, whereas the two-dimensional code can only do it every two symbols. Depending on the application, this advantage is likely to be superseded by the redundancy advantage of the two-dimensional code. Also as in the one-dimensional case, there are certain complexity advantages, in both encoding and adaptation when using the subset of parameters of the form $k = 2^r$. In this case, an adaptation strategy that estimates the best parameter r directly from the statistic S_t, without the need to compare it with precomputed crossover points, can be derived for the codes C_k, as was done in [3] and [13] for two-sided geometric distributions. We omit the details, since both the technique and the resulting parameter estimation method are similar to those in the references.

Figure 7 presents plots of redundancy for various code families as a function of q, measured in bits per integer symbol relative to the entropy of the geometric distribution (recall that the latter is given by $H(q) = \frac{h(q)}{1-q}$, where $h(q)$ is the binary entropy function [2]). Plots are shown for the optimal prefix code for each value of q (estimated numerically over a dense grid of values of q, and in sufficient precision to make the estimation error smaller than the plot resolution), the best Golomb code, the best code C_{-k} or

C_k for each q, and the limit code $C_{-\infty}$. In the figure, we can observe the advantage in redundancy for the codes C_{-k} (or $C_{-\infty}$) and C_k over Golomb codes, except in the region where the best codes of both types are equivalent (i.e., the optimality regions of C_1 and C_2). The redundancy advantage is near $2:1$ (as expected) at the limit of $q \to 0$ and it peaks near $q = 0.28$ (at more than $13.6:1$). A redundancy advantage close to $2:1$ is observed also as $q \to 1$. The advantage of C_k over symbol-by-symbol Golomb codes is consistent with Corollary 1, and, in fact, the plot in Figure 7 can be regarded as "visual evidence" for the corollary.

The asymptotic behavior of the redundancy of C_k in the regime $q \to 1$, shown in more detail in Figure 8, is oscillatory, as is also the case for Golomb codes [2]. The limiting behavior of the redundancy can be characterized precisely, as we show next.

Corollary 6: Let $\lambda_k = 2^M/k^2$, where M is as defined in Theorem 4. As $k \to \infty$, the redundancy of the code C_k at $q = 2^{-1/k}$ is

$$R(k) = \frac{1}{2}\,(1 + \log\lambda_k) + 2^{1-2\sqrt{\lambda_k - \frac{1}{2}}}\left(1 + \frac{2}{\log e}\sqrt{\lambda_k - \frac{1}{2}}\right) - \log(e\log e) + o(1)\,. \tag{43}$$

Remark. We have $\frac{3}{4} \lessapprox \lambda_k \lessapprox \frac{3}{2}$, where $\lessapprox$ denotes inequality up to asymptotically negligible terms. For large k, as k increases, λ_k sweeps its range decreasing from $\frac{3}{2}$ to $\frac{3}{4}$, at which point M_k increases by one, and λ_k resets to $\frac{3}{2}$, starting a new cycle.

Proof of Corollary 6: We derive, from (26), an asymptotic expression for the code length $\mathcal{L}_q(C_k)$. To estimate the parameter j in (26), we need to solve the quadratic equation $\Delta(x) = 0$, with $\Delta(x)$ as defined in Theorem 4. Writing $2^M = \lambda_k k^2$, it is readily verified that the largest solution to the equation is $\xi = \left(2\sqrt{\lambda_k - \frac{1}{2}} - 1\right) k + O(1) \triangleq \alpha\, k + O(1)$. Thus, $j = \alpha\, k + O(1)$, and $q^j = 2^{-\alpha} + O(k^{-1})$. Writing also $q = 2^{-1/k} = 1 - \frac{\ln 2}{k} + O(k^{-2})$, and noting that $\Delta(j) = O(k)$, we obtain, from (26),

$$\mathcal{L}_q(C_k) = M + 1 + 2^{1-\alpha}\big(1 + (1+\alpha)\ln 2\big) + o(1)\,.$$

As for the entropy, we have

$$H(q) = \frac{-q\log q}{1-q} - \log(1-q) = \log(e\log e) + \log k + o(1) = \log(e\log e) + \frac{1}{2}\,(M - \log\lambda_k) + o(1)\,.$$

The claimed result (43) follows by substituting the asymptotic expressions for $\mathcal{L}_q(C_k)$ and $H(q)$ in the formula for the redundancy per symbol, namely, $R(k) = \frac{1}{2}\mathcal{L}_q(C_k) - H(q)$. ■

The limits of oscillation of the function R_k can be obtained by numerical computation, yielding $R_1 \triangleq \liminf_{k\to\infty} R(k) = 0.014159\ldots$ and $R_2 \triangleq \limsup_{k\to\infty} R(k) = 0.014583\ldots$. These limits are shown in Figure 8. The corresponding limits for the redundancy of the Golomb codes are, respectively, $R'_1 = 0.025101\ldots$ and $R'_2 = 0.032734\ldots$ [2].

Corollary 6 applies to the discrete sequence of redundancy values at the points $q = 2^{-1/k}$. It is not difficult to prove that the same behavior, and in particular the limits R_1 and R_2, apply also to the continuous redundancy curve obtained when using the best code C_k at each arbitrary value of q. This follows from the readily verifiable fact that as q varies in the interval $2^{-1/k} \le q \le 2^{-1/(k+1)}$, the maximal

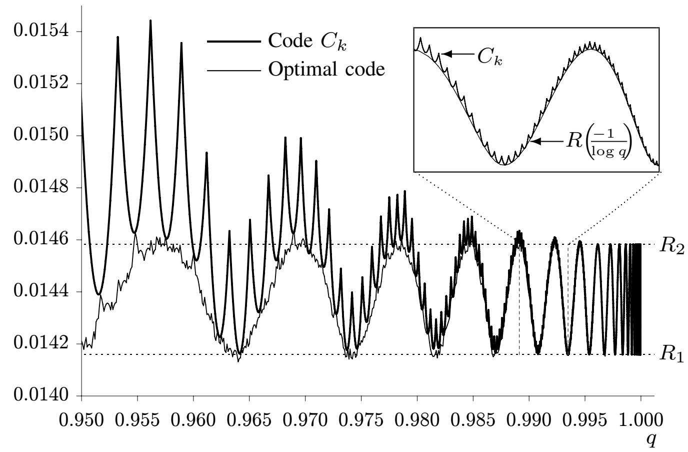


Fig. 8. Redundancy as $q \to 1$ ($k \to \infty$). Dashed lines show the asymptotic limits R_1 and R_2. The inset closes up further on a narrow segment, showing the redundancy of the codes C_k vs. the asymptotic estimate (43).

variation in both the code length under C_k and the distribution entropy is bounded by $O(k^{-1})$. Figure 8 suggests that the same oscillatory behavior might apply also to the redundancy curve of the optimal prefix code for each value of q. It follows from the foregoing discussion that this is true for the limit superior R_2. The question remains open, however, for the limit inferior R_1, which is an upper bound for the limit inferior of the optimal redundancy.

Appendix A
Proofs for Subsection IV-B

We recall that we consider a 4-uniform probability distribution $\mathbf{p} = (p_1, p_2, \ldots, p_N)$, where probabilities are listed in non-increasing order, and an optimal tree T for $\mathbf{p}$, with $f_T \leq 2$. We define $m = \lceil \log N \rceil$, and we denote by n_ℓ the number of leaves at depth ℓ in T.

Proof of Lemma 6: Say T has $t > 0$ leaves at depths $\ell < m-2$. Then, T has no leaves at depths $\ell' \geq m$, and it can have a total of at most $2^{m-1} - 3t$ leaves altogether. But $N > 2^{m-1}$, a contradiction. Say now that T has nodes at depth $m+2$. Then all of its leaves must be at depths $\ell' \geq m$, and some must be at depths strictly greater than m. Thus, T, being full, must have more than $2^m \geq N$ leaves, again a contradiction. The second claim of the lemma is a straightforward consequence of $f_T \leq 2$. ■

Proof of Lemma 7: Let $\mathbf{N}_T = (n_{M-1}, n_M, n_{M+1})$ be the compact profile of a tree T with N leaves and $f_T \leq 2$. Clearly, n_{M+1} must be even, and we write $n_{M+1} = 2c$ for some nonnegative integer c. The components of $\mathbf{N}_T$ must satisfy

$$n_{M-1} + n_M + 2c = N \,. \tag{44}$$

By Kraft's equality, which must hold for the full tree T, we have

$$4n_{M-1} + 2n_M + 2c = 2^{M+1} \,, \tag{45}$$

which holds also in the case $c = 0$. From (44) and (45), we obtain

$$n_{M-1} = 2^M - N + c\,. \tag{46}$$

Now, from (46) and (44), we obtain

$$n_M = 2N - 2^M - 3c\,. \tag{47}$$

Equations (46) and (47), together with the definition of c yield the profile (12). The valid range of variation of c is determined by the non-negativity constraints on the entries of the profile. When $M = m-1$ ($\sigma = 1$), the lower limit $\underline{c}_\sigma = N - 2^{m-1}$ is determined by the nonnegativity of n_{M-1}. Since $2^M \geq N$ when $M = m$, the lower limit is the trivial $\underline{c}_0 = 0$ in this case. In both cases, the upper limit $\overline{c}_\sigma = \lceil \frac{2N-2^M}{3} \rceil$ is determined by the nonnegativity of n_M. ■

Proof of Lemma 8: For a given value of $\sigma \in \{0, 1\}$, assume c and c' are indices such that $\underline{c}_\sigma < c' \leq c \leq \overline{c}_\sigma$, and let $\mathbf{s}_\sigma$ be the segment of $\mathbf{s}$ corresponding to σ. By (14) and the monotonicity of the weights, we have

$$D_{\sigma,c'} = p_{N-2c'+1} + p_{N-2c'+2} - p_{2^M-N+c'} \leq p_{N-2c+1} + p_{N-2c+2} - p_{2^M-N+c} = D_{\sigma,c}\,.$$

Thus, if $D_{\sigma,c} < 0$ then $D_{\sigma,c'} < 0$, and if $D_{\sigma,c} = 0$ then $D_{\sigma,c'} \leq 0$. It follows that $\mathbf{s}_\sigma$ is non-decreasing. It remains to prove that $-\mathrm{sg}(D_{1,\underline{c}_1+1}) \leq \mathrm{sg}(D_{0,1})$. Assume that $D_{0,1} \leq 0$. Then, we have

$$\begin{aligned} D_{1,\underline{c}_1+1} &= p_{2^m-N-1} + p_{2^m-N} - p_1 \geq 2p_{2^m-N+1} - p_1 \\ &\geq 2\left(p_{N-1} + p_N\right) - p_1 \geq 4p_N - p_1 \geq 0\,, \end{aligned}$$

where the equality follows from (14) and the definition of $\underline{c}_1$, the first and third inequalities from the monotonicity of $\mathbf{p}$, the second inequality from our assumption on $D_{0,1}$, and the last inequality from the 4-uniformity of $\mathbf{p}$. Hence, we must have $D_{1,\underline{c}_1+1} \geq 0$. Similarly, if $D_{0,1} < 0$, then we must have $D_{1,\underline{c}_1+1} > 0$. Therefore, $-\mathrm{sg}\left(D_{1,\underline{c}_1+1}\right) \leq \mathrm{sg}(D_{0,1})$, as claimed. ■

Proof of Theorem 3: The theorem follows directly from Lemma 8, observing also that by the assumptions of the theorem, and by Lemma 7, at least one of the trees $T_{\sigma,c}$, $(1, \overline{c}_1) \preceq (\sigma, c) \preceq (0, \overline{c}_0)$ must be optimal for $\mathbf{p}$. ■

Appendix B
Proofs for Subsection IV-C

We derive the proof of Theorem 4 through a series of lemmas. We recall that we seek an optimal tree for the source $\hat{\mathcal{A}}_k$ of (10), with vector of (unnormalized) weights

$$\mathbf{p} = (q^0, q^1, q^1, \ldots, q^j, q^j, \ldots, q^j, \ldots, q^{2k-3}, q^{2k-3}, q^{2k-2}),$$

with $q = 2^{-1/k}$, and where q^j is repeated $j+1$ times for $0 \leq j \leq k-1$, and $2k-1-j$ times for $k \leq j \leq 2k-2$. For succinctness, in this appendix, when we say "optimal" we mean "optimal for $\hat{\mathcal{A}}_k$." Notice that, in $\mathbf{p}$, three consecutive weights are never distinct; we refer to this fact as the "three consecutive weights" property. Throughout the appendix, we assume that $k > 2$, as we recall that optimal trees for $k = 1, 2$ are fully characterized in Remark 2 following Theorem 2.

Lemma 12: Trees $T_{\sigma,c}$ with $c = \overline{c}_\sigma$ are not optimal. Consequently, the profile (n_{M-1}, n_M, n_{M+1}) of an optimal tree has $n_M \geq 3$.

Proof: Recalling the profile $\mathbf{N}_{T_{\sigma,c}}$ in (12), with $c = \overline{c}_\sigma$ and $k > 2$, we have $n_M \in \{0, 1, 2\}$, $n_{M-1} \geq 1$ and $n_{M+1} \geq 2$. Let q^ℓ be the lightest weight on level $M - 1$. By the "three consecutive weights" property, the two heaviest weights on level $M + 1$ are greater than or equal to $q^{\ell+2}$. Recalling the expression for $D_{\sigma,c}$ in (14), and the interpretation that follows it, we obtain $D_{\sigma,\overline{c}_\sigma} \geq q^\ell(1-2q^2) > 0$. Thus, by Theorem 3, $T_{\overline{c}_\sigma}$ is not optimal. An optimal tree $T_{\sigma,c}$ would, therefore, have $c < \overline{c}_\sigma$, and, thus, $n_M \geq 3$. ■

The following lemma gives a first, rough approximation of the distribution of weights by levels in an optimal tree $T_{\sigma,c}$, which will allow us to identify the appropriate range (i.e., (16) or (17)) for the heaviest and the lightest weights on level M of the tree.

Lemma 13: Let $T_{\sigma,c}$ be an optimal tree, and let q^j and $q^{2k-2-j'}$ denote, respectively, the heaviest and the lightest weights on level M of the tree. Then, we have $j \leq k - 1$, $j' \leq k - 1$, and $j + j' \leq k$.

Proof: Consider first the case where $c > \underline{c}_\sigma$, i.e., all the components of the profile $\mathbf{N}_{T_{\sigma,c}}$ are positive. The lightest weight on level $M - 1$ of the tree immediately precedes q^j in $\mathbf{p}$. Hence, it is of the form $q^{j-\varepsilon}$, with $\varepsilon \in \{0, 1\}$. On the other hand, reasoning similarly, the heaviest two weights on level $M + 1$ are of the form $q^{2k-2-j'+\varepsilon'}$ and $q^{2k-2-j'+\varepsilon'+\varepsilon''}$, where $\varepsilon', \varepsilon'' \in \{0, 1\}$ and $\varepsilon' + \varepsilon'' \leq 1$ (due to the "three consecutive weights" property). Since $T_{\sigma,c}$ is optimal, by the definition of $D_{\sigma,c}$ in (13), we must have $D_{\sigma,c} \leq 0$. Applying (14), the above constraints on $\varepsilon, \varepsilon', \varepsilon''$, and the fact that $q^k = \frac{1}{2}$, we get

$$0 \geq D_{\sigma,c} = -q^{j-\epsilon} + q^{2k-2-j'+\epsilon'} + q^{2k-2-j'+\epsilon'+\epsilon''} \geq -q^{j-1} + 2q^{2k-1-j'} = -q^{j-1} + q^{k-1-j'} .$$

Thus, $j + j' \leq k$. Since both j and j' are positive when $c > \underline{c}_\sigma$, the claim of the lemma follows in this case.

Consider now the case where $c = \underline{c}_\sigma$, i.e., $T_{\sigma,c}$ is a quasi-uniform tree. If $\sigma = 0$, we have $n_{M+1} = 0$, and, thus, the lightest weight on level M is $p_{k^2} = q^{2k-2}$, and $j' = 0$. For the heaviest weight on level M, we have $p_{2^m-k^2+1} = q^j$. By (18), we have $2^m - k^2 + 1 \leq k(k + 1)/2$. Recalling the order and structure of $\mathbf{p}$, we obtain $q^j = p_{2^m-k^2+1} \geq p_{k(k+1)/2} = q^{k-1}$. Thus, $j \leq k - 1$. The case of $c = \underline{c}_\sigma$ and $\sigma = 1$ is argued similarly, using (20) in lieu of (18), and leading to $j = 0$ and $j' \leq k - 1$. ■

It follows from Lemma 13 that in an optimal tree, the heaviest weight on level M is covered by (16) in Lemma 9 (and, thus, so is any weight on level $M-1$), while the lightest weight on level M is covered by (17) in that lemma (and, thus, so is any weight on level $M + 1$). Consequently, an optimal tree is completely determined by a tuple $\mathbf{j} = (j, r, j', r')$, with $0 \leq j, j' \leq k - 1$, $0 \leq r \leq j$, and $0 \leq r' \leq j'$.

The profile of the tree is then given by

$$
\begin{aligned}
n_{M-1} &= \frac{j(j+1)}{2} + r\,, && (48)\\
n_{M+1} &= \frac{j'(j'+1)}{2} + r'\,, && (49)\\
n_M &= k^2 - n_{M-1} - n_{M+1}\,. && (50)
\end{aligned}
$$

The following lemma presents a characterization of the least value of c for which $T_{\sigma,c}$ is optimal. The lemma follows immediately from Theorem 3 and Lemma 10.

Lemma 14: Let c_k be the least value of c such that $T_{\sigma,c}$ is optimal. Then, either $D_{\sigma,\underline{c}_\sigma+1} \geq 0$ (with $c_k = \underline{c}_\sigma$), or $D_{\sigma,c_k} < 0$ and $D_{\sigma,c_k+1} \geq 0$ (with $c_k > \underline{c}_\sigma$).

Define the function

$$F(j,r,j',r') = 2k^2 - 2^{M+1} + j(j+1) + 2r - \frac{j'(j'+1)}{2} - r'\,, \tag{51}$$

acting on tuples $\mathbf{j} = (j,r,j',r')$ for a given value of k. Next, we derive a set of conditions on the tuple $\mathbf{j}$ corresponding to the tree T_{σ,c_k} characterized in Lemma 14.

Lemma 15: Let $\mathbf{j} = (j,r,j',r')$ be the tuple defining the profile of T_{σ,c_k} in (48)–(50). Then,

$$F(j,r,j',r') = 0, \tag{52}$$

and exactly one of the following conditions holds:

(i) $j,j' > 0$, $j + j' = k-2$. Either $r = 0$ and $0 \leq r' \leq j'$, or $1 \leq r \leq j$ and $r' \in \{0,1\}$.

(ii) $j,j' > 0$, $j+j' = k-1$, $r = 0$ and $r' \in \{0,1\}$.

(iii) $j' = 0$, $r' = 0$, $j \in \{k-2, k-1\}$, $0 \leq r \leq j$.

(iv) $j = 0$, $r = 0$, $j' \in \{k-2, k-1\}$, $0 \leq r' \leq j'$.

Conversely, if $\mathbf{j} = (j,r,j',r')$ satisfies (52) and one of the conditions (i)–(iv), then $\mathbf{j}$ defines T_{σ,c_k}.

Proof: The necessity of (52) follows from the definition of $F(j,r,j',r')$ and from (46), setting $c = \frac{1}{2}n_{M+1}$, substituting the expressions from (48) and (49) for n_{M-1} and n_{M+1}, respectively, and rearranging terms. In fact, (52) must hold for any optimal tree, not just for $c = c_k$. Conditions (i)–(iv) will follow from an exhaustive case study of configurations that yield the inequalities on the quantities $D_{\sigma,c}$ that characterize the point $c = c_k$, as stated in Lemma 14.

Consider, first, the case where $c_k > \underline{c}_\sigma$. Then, for $c = c_k$, by Lemma 14, we have $D_{\sigma,c} < 0$ and $D_{\sigma,c+1} \geq 0$. Writing down the expressions for $D_{\sigma,c}$ and $D_{\sigma,c+1}$ explicitly according to (14), we observe that six weights are involved, as illustrated in Figure 9. In order to switch from a negative $D_{\sigma,c}$ to a nonnegative $D_{\sigma,c+1}$, we must have a decrease from $p_{2^M-k^2+c}$ to $p_{2^M-k^2+c+1}$, or an increase from $p_{k^2-2c+1}+p_{k^2-2c+2}$ to $p_{k^2-2c-1}+p_{k^2-2c}$, or both. By the definitions of j and j', we have $p_{2^M-k^2+c+1} = q^j$, and $p_{k^2-2c} = q^{2k-2-j'}$. Taking into account that consecutive weights can vary at most by a factor of

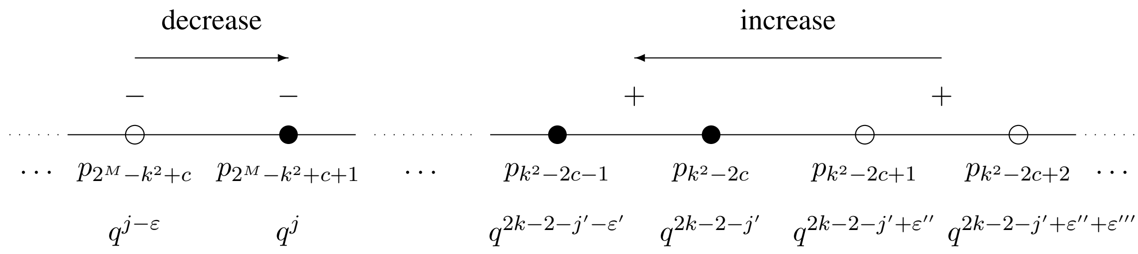


Fig. 9. Weights involved in the conditions for $c = c_k$: $\circ$ weights in $D_{\sigma,c}$, $\bullet$ weights in $D_{\sigma,c+1}$.

TABLE V

THE POSSIBLE CASES FOR $(\varepsilon, \varepsilon', \varepsilon'', \varepsilon''')$ FROM (53)–(54), AND THE CONDITIONS IMPOSED ON (j, r, j', r') AT $c = c_k$.

$\boldsymbol{\varepsilon} = (\varepsilon, \varepsilon', \varepsilon'', \varepsilon''')$	Conditions on (j, r, j', r')
(1,0,0,0)	$j + j' = k - 2,\ r = 0,\ 2 \le r' \le j' - 1$
(1,0,0,1)	$j + j' \in \{k-2, k-1\},\ r = 0,\ r' = 1$
(1,0,1,0)	$j + j' \in \{k-2, k-1\},\ r = 0,\ r' = 0$
(0,0,0,1)	$j + j' = k - 2,\ 1 \le r \le j,\ r' = 1$
(0,0,1,0)	$j + j' = k - 2,\ 1 \le r \le j,\quad r' = 0$
(1,1,0,0)	$j + j' = k - 2,\ r = 0,\ r' = j'$
(1,1,0,1)	$j + j' \in \{k-1, k-2\},\ r = 0,\ r' = j' = 1$
(0,1,0,0)	no solutions j, j'; case cannot occur at $c = c_k$
(0,1,0,1)	$j + j' = k - 2,\ 1 \le r \le j,\ r' = j' = 1$

q, we can write, for the other weights involved,

$$p_{2^M-k^2+c} \;=\; q^{j-\varepsilon}, \tag{53}$$

$$p_{k^2-2c-1} \;=\; q^{2k-2-j'-\varepsilon'},\ p_{k^2-2c+1} \;=\; q^{2k-2-j'+\varepsilon''},\ p_{k^2-2c+2} \;=\; q^{2k-2-j'+\varepsilon''+\varepsilon'''}, \tag{54}$$

where $\varepsilon, \varepsilon', \varepsilon'', \varepsilon''' \in \{0, 1\}$, and, due to the "three consecutive weights" property, we must have $\varepsilon' + \varepsilon'' \le 1$ and $\varepsilon'' + \varepsilon''' \le 1$. Table V summarizes the patterns of values of $\boldsymbol{\varepsilon} = (\varepsilon, \varepsilon', \varepsilon'', \varepsilon''')$ that satisfy these constraints and also produce the combination of weight increases or decreases necessary to satisfy the conditions for $c = c_k$. On the right column of the table, we list the conditions imposed on $\mathbf{j}$ by the constraints of each case. To illustrate the proof approach, we derive these conditions, below, for the representative case $\boldsymbol{\varepsilon} = (1, 0, 0, 1)$. The other cases follow using similar arguments, which are also similar to those used in the proof of Lemma 13 (here, more parameters are assumed known, which allows us to obtain tighter bounds).

Assume $\boldsymbol{\varepsilon} = (1, 0, 0, 1)$. Then, writing the conditions on $D_{\sigma,c}$ and $D_{\sigma,c+1}$ at $c = c_k$ explicitly, substituting for the weights using the known values in $\boldsymbol{\varepsilon}$, and recalling that $q^k = \frac{1}{2}$, we obtain

$$\begin{aligned} 0 \;&>\; D_{\sigma,c} = p_{k^2-2c+1} + p_{k^2-2c+2} - p_{2^M-k^2+c} = q^{2k-2-j'} + q^{2k-1-j'} - q^{j-1} \\ &>\; 2q^{2k-1-j'} - q^{j-1} = q^{k-1-j'} - q^{j-1}\,, \end{aligned} \tag{55}$$

and

$$
\begin{aligned}
0 \quad &\leq \quad D_{\sigma,c+1} = p_{k^2-2c-1} + p_{k^2-2c} - p_{2^M-k^2+c+1} = q^{2k-2-j'} + q^{2k-2-j'} - q^j \\
&= \quad 2q^{2k-2-j'} - q^j = q^{k-2-j'} - q^j \,.
\end{aligned}
\tag{56}
$$

It follows that $k-2 \leq j+j' \leq k-1$, as claimed in the second row of Table V. The conditions on r and r' follow from Lemma 9, observing that r resets to zero at points where j increases, and similarly with r' relative to j'. In this case, $p_{2^M-k^2+c}$ is the last weight of the form q^{j-1}, and, thus, we have $n_{M-1} = 2^M - k^2 + c = j(j+1)/2$ and $r = 0$; scanning $\mathbf{p}$ from right to left, p_{k^2-2c+2} is the last weight of the form $q^{2k-1-j'}$, and, thus, we have $n_{M+1} = 2c = j'(j'+1)/2 + 1$, and $r' = 1$.

It is readily verified that all the cases on the right column of Table V satisfy either Condition (i) or Condition (ii) of the lemma.

Consider now the case where $c_k = \underline{c}_\sigma$. In this case, the tree is quasi-uniform. When $\sigma_k = 0$, since $n_{M+1} = 0$, we have $j' = r' = 0$. The condition $j \leq k-1$ was established in Lemma 13, while the condition $j \geq k-2$ follows directly from $D_{\sigma,\underline{c}_\sigma+1} = D_{\sigma,1} \geq 0$. Thus, Condition (iii) of the lemma is satisfied in this case. Similarly, when $c_k = \underline{c}_\sigma$ and $\sigma_k = 1$, we have $j = r = 0$, $j' \leq k-1$ was established in Lemma 13, and $j' \geq k-2$ follows from $D_{\sigma,\underline{c}_\sigma} \geq 0$. Thus, Condition (iv) of the lemma is satisfied in this case.

To prove the sufficiency of the conditions of the lemma, we first claim that, with $\mathbf{j}$ satisfying the conditions, the profile $\mathbf{N} = (n_{M-1}, n_M, n_{M+1})$ defined in (48)–(50) defines a valid tree. Clearly, n_{M-1} and n_{M+1} are non-negative. To verify that n_M is also non-negative, we write

$$
n_{M-1} + n_{M+1} = \frac{j(j+1)}{2} + \frac{j'(j'+1)}{2} + r + r' < \frac{(j+j'+1)^2}{2} + j + j',
$$

where the inequality follows from the fact that $(a+b+1)^2 > a(a+1)+b(b+1)$ for $a, b \geq 0$, and from the inequalities $r \leq j$ and $r' \leq j'$. With $j+j' \leq k-1$, it follows that $n_{M-1} + n_{M+1} < k-1+k^2/2 < k^2$. Hence, n_M, as defined in (50), is positive. On the other hand, (52), together with the fact that the components of $\mathbf{N}$ add up to k^2, is equivalent to the Kraft equality for $\mathbf{N}$. Therefore, $\mathbf{N}$ defines a valid tree $T_{\sigma,c}$. It is readily verified that if either Condition (i) or (ii) is satisfied, then the parameters (σ, c) of $T_{\sigma,c}$ satisfy $c > \underline{c}_\sigma$, $D_{\sigma,c} < 0$, and $D_{\sigma,c+1} \leq 0$. Thus, by Lemma 8, we have $c = c_k$. Similarly, if either Condition (iii) or (iv) is satisfied, we have $c = \underline{c}_\sigma$, $D_{\sigma,\underline{c}_\sigma+1} \geq 0$, and, again, $c = c_k$. ■

The following lemma explores some properties of the function $\Delta(x)$ defined in (21).

Lemma 16: (i) For any x, we have $\Delta(x+1) = \Delta(x) + x + k\,$.

(ii) We have $\Delta(-1) \leq 0$ and $\Delta(k) > 0$. Thus, x_0, the largest real root of Δ, satisfies $-1 \leq x_0 < k$.

(iii) The values $\Delta(k-1)$ and $\Delta(k-2)$ are even integers.

Proof: (i) The claim is readily verified by direct application of (21).

(ii) Setting $x = -1$ in (21), and recalling that $Q = k^2 - \lceil k(k-1)/4 \rceil$ and $M = \lceil \log Q \rceil$, we obtain

$$
\Delta(-1) = 2(k^2 - \frac{k(k-1)}{4} - 2^M) = 2\big(Q - 2^M + \frac{1}{2}\mathbf{1}_{(k \bmod 4) \in \{2,3\}}\big) = \mathbf{1}_{(k \bmod 4) \in \{2,3\}} + 2(Q - 2^M),
$$

where $\mathbf{1}_{\mathcal{P}} = 1$ if the predicate $\mathcal{P}$ is true, or $\mathbf{1}_{\mathcal{P}} = 0$ otherwise. It follows that $\Delta(-1)$ can be positive only if $(k \bmod 4) \in \{2,3\}$ and $Q = 2^M$. Writing $Q = Q(k)$, and computing explicitly $Q(4\ell+2) = (4\ell+3)(3\ell+1)$ and $Q(4\ell+3) = (\ell+1)(12\ell+7)$, we conclude that Q has at least one odd divisor when $(k \bmod 4) \in \{2,3\}$. Therefore, we must have $\Delta(-1) \leq 0$.

Furthermore, since $Q \leq 2^M \leq 2Q-1$, we have

$$\begin{aligned}\Delta(k) &= 2k^2 - 2^{M+1} + k(k+1) - 1 \geq 2k^2 - 4Q + k(k+1) + 1 \\ &= -2k^2 + 4\left\lceil \frac{k(k-1)}{4} \right\rceil + k(k+1) + 1 \geq -2k^2 + k(k-1) + k(k+1) + 1 = 1.\end{aligned}$$

Thus, $\Delta(k) > 0$, and, since the coefficient of x^2 in $\Delta(x)$ is $\frac{1}{2}$, x_0 must be in the claimed range.

(iii) By direct computation, we have $\Delta(k-1) = 2\,k^2 - 2^{M+1} + (k-1)k$ and $\Delta(k-2) = 2\,k^2 - 2^{M+1} + (k-2)(k-1)$. Since $k > 2$ and $M > 0$, both values are even. ■

To complete the proof of Theorem 4, we will construct a tuple $\mathbf{j} = (j, r, j', r')$ that satisfies the conditions of Lemma 15, and, thus, defines the sought parameter pair (σ_k, c_k).

Proof of Theorem 4: It follows immediately from the definition of $\Delta(x)$ in (21) and of $F(j, r, j', r')$ in (51) that for j, r, j', r' we have

$$F(j, r, j', r') = \Delta(j) + \frac{(k-j-2)(k-j-1)}{2} - \frac{j'(j'+1)}{2} + 2r - r'\,. \tag{57}$$

When $j' = k-j-2$, this reduces to

$$F(j, r, j', r') = \Delta(j) + 2r - r'\,, \tag{58}$$

while with $j' = k-1-j$ we get

$$F(j, r, j', r') = \Delta(j) + 2r - r' - (k-j-1)\,. \tag{59}$$

We will use these relations to verify that the solutions constructed below satisfy (52). Let x_0 be the largest real root of $\Delta(x)$, and let $\xi = \lfloor x_0 \rfloor$. By Lemma 16(ii), we have $-1 \leq \xi < k$, $\Delta(\xi) \leq 0$, and $\Delta(\xi+1) > 0$. We consider three main cases for $\Delta(\xi)$, and for each case (and possible sub-cases) we define a tuple $\mathbf{j} = (j, r, j', r')$ and verify that it satisfies the conditions of Lemma 15.

1) $0 \leq -\Delta(\xi) \leq 2\xi:$ Let $j = \xi$, $r = \lfloor \frac{-\Delta(j)+1}{2} \rfloor$ and $r' = -\Delta(j) \mod 2$. By the assumptions of the case on $\Delta(\xi)$, we have $j \geq 0$. As for j', we have the sub-cases below. At the end of each sub-case, we note which of Conditions (i)–(iv) of Lemma 15 is satisfied.
   a) $j = 0:$ We must have $\Delta(0) = 0$, so we get $r = r' = 0$, and we set $j' = k-2$ (Condition (iv)).
   b) $j \in \{k-2, k-1\}:$ By Lemma 16(iii), $\Delta(j)$ is even, and $r' = 0$. We get $r = -\frac{\Delta(j)}{2}$ and $0 \leq r \leq j$ by the assumptions on $\Delta(\xi)$, and we set $j' = 0$ (Condition (iii)).
   c) $0 < j < k-2:$ Set $j' = k-2-j$. From the choices for r and r', we get $0 \leq r \leq j$ and $0 \leq r' \leq 1 \leq j'$ (Condition (i)).

   To verify that (52) is satisfied, we apply (58) for sub-cases a) and c), and for sub-case b) with

$j = k-2$. We apply (59) for sub-case b) with $j = k-1$. For example, for sub-case c), by (58) and the definitions of r and r', we have,

$$F(j,r,j',r') = \Delta(j) + 2r - r' = \Delta(j) + 2\left\lfloor \frac{1-\Delta(j)}{2} \right\rfloor - r' = \Delta(j) + 2\,\frac{r'-\Delta(j)}{2} - r' = 0\,.$$

Verification of $F = 0$ for the other sub-cases follows along similar lines.

2) $-\Delta(\xi) \in \{2\xi+1,\, 2\xi+2\}$: Let $j = \xi+1$. By Lemma 16(ii), we have $0 \le j \le k$. We claim that $j \le k-1$. Assume, contrary to the claim, that $j = k$. Then, $-\Delta(k-1) = -\Delta(\xi) = 2k - \varepsilon$ with $\varepsilon \in \{0,1\}$, and, by Lemma 16(i), we have $\Delta(\xi+1) = \Delta(k) = \Delta(k-1) + 2k - 1 = \varepsilon - 1 \le 0$, contradicting Lemma 16(ii), which establishes $\Delta(\xi+1) > 0$. Thus, we have $0 \le j \le k-1$, and, defining $j' = k-1-j$, we also have $0 \le j' \le k-1$. By Lemma 16(i), we have $\Delta(j) = \Delta(\xi+1) = \Delta(\xi)+\xi+k$, and, by the conditions of the case on $\Delta(\xi)$, we get $\Delta(j) \in \{k-j, k-j-1\}$. Define $r = 0$, and $r' = \Delta(j) - (k-j-1)$, which implies $r' \in \{0,1\}$. Thus, whenever $0 < j < k-1$, $\mathbf{j} = (j,r,j',r')$ satisfies Condition (ii) of Lemma 15. When $j = 0$, $\mathbf{j}$ satisfies Condition (iv), and when $j = k-1$, it satisfies Condition (iii) as long as $r' = 0$. We claim that when $r' = 1$, we must have $j < k-1$. Otherwise, if $r' = 1$ and $j = k-1$, then, by the definition of r', we have $\Delta(k-1) = \Delta(j) = r' + (k-j-1) = 1$, contradicting Lemma 16(iii). Thus, $\mathbf{j}$ satisfies one of the conditions (ii)–(iv) of Lemma 15. By (59) and the definitions of r and r', $\mathbf{j}$ also satisfies (52).

3) $-\Delta(\xi) \ge 2\xi + 3$: Let $j = \xi+1$. By Lemma 16(ii), we have $0 \le j \le k$. We claim that $j \le k-2$. Assume, contrary to the claim, that $j = k-1$. Then, $\xi = k-2$, and, by the assumptions of the case, we have $-\Delta(k-2) \ge 2(k-2)+3 = 2k-1$. Applying Lemma 16(i), we get $\Delta(\xi+1) = \Delta(k-1) = \Delta(k-2) + (k-2) + k = \Delta(k-2) + 2k - 2 \le -1$, contradicting Lemma 16(ii), since we must have $\Delta(\xi+1) > 0$. Similarly, if $j = k$, then $-\Delta(k-1) \ge 2k+1$ and $\Delta(k) = \Delta(k-1) + 2k - 1 \le -2$, again contradicting Lemma 16(ii). Thus, we have $0 \le j \le k-2$, and we can define $j' = k-2-j$, which also satisfies $0 \le j' \le k-2$. By Lemma 16(i), and the conditions of the case on $\Delta(\xi)$, we have $\Delta(j) = \Delta(\xi+1) = \Delta(\xi)+\xi+k \le k-\xi-3 = k-2-j = j'$. Define $r = 0$, and $r' = \Delta(j)$, satisfying $0 \le r' \le j'$. Thus, $\mathbf{j} = (j,r,j',r')$ satisfies Condition (i) of Lemma 15. By (58) and the definitions of r and r', $\mathbf{j}$ also satisfies (52).

Cases 1–3 above cover all possible values of $\Delta(\xi)$, and in all cases, we have exhibited an explicit tuple $\mathbf{j} = (j,r,j',r')$ satisfying the conditions of Lemma 15, and, therefore, defining the optimal tree T_{σ_k, c_k}. It can readily be verified that the definitions of j and r in (22)–(23) summarize the corresponding definitions in the cases of the proof, with (22) corresponding to Case 1, and (22) to Cases 2 and 3. Furthermore, the definition of c_k in (24) reflects the parameter $c = n_{M-1} - 2^M + k^2$ in the profile (48)–(50) defined by $\mathbf{j}$ for $c = c_k$. ■

Proof of Corollary 1: By the structure of C_k in Theorem 2, it suffices to prove that $Q_k \cdot Q_k$ is not optimal for the finite source $\mathcal{A}_k$. Let $h = \lceil \log k \rceil$ and $a = 2^h - k$, with $0 \le a < 2^{h-1}$. From the profile of Q_k given in in Section II-B, one derives the profile of $Q_k \cdot Q_k$, obtaining

$$\mathbf{N}_{Q_k \cdot Q_k} = \left(n_{2h-2},\, n_{2h-1},\, n_{2h}\right) = \left(a^2,\, 2a(k-a),\, (k-a)^2\right)\,.$$

Since $Q_k \cdot Q_k$ has fringe thickness $f_T \leq 2$, it has a representation T_{σ_g, c_g}, for some parameters σ_g, c_g, as defined in Lemma 7, with $N = k^2$. The case $a = 0$ (i.e., $k = 2^h$) is readily discarded as sub-optimal for $k > 2$, as it corresponds to a uniform tree with 2^{2h} leaves, which cannot be optimal for $\hat{\mathcal{A}}_k$ since $p_{k^2} + p_{k^2-1} < p_1$ for that source. Also, we can assume that σ_g is such that Lemma 10 is satisfied, and that n_{2h-2} and n_{2h} are such that they can be written, respectively, as n_{M-1} and n_{M+1} in (48)–(49), with j and j' satisfying Lemma 13. Otherwise, T_{σ_g, c_g} is not optimal, and the corollary is proved. By Lemma 9, we can write $a^2 < \frac{1}{2}(j+1)(j+2) < \frac{1}{2}(j+2)^2$, or $j > \sqrt{2}\, a - 2$. Similarly, we have $(k-a)^2 < \frac{1}{2}(j'+1)(j'+2) < \frac{1}{2}(j'+2)^2$, or $j' > \sqrt{2}(k-a) - 2$. Adding up, we obtain $j + j' > \sqrt{2}\, k - 4$, and, hence, for $k \geq 10$, $j + j' > k$, contradicting Lemma 13. For the remaining cases, if $k \in \{7, 9\}$ one verifies that σ_g violates Lemma 10, and for $k \in \{3, 5, 6\}$, one can easily verify, by direct inspection, that T_{σ_g, c_g} is sub-optimal for $\hat{\mathcal{A}}_k$. ■

APPENDIX C
LAYER TRANSITIONS IN THE CODES C_{-k}

In each layer transition described below, we assume that we start from a layer $\mathbf{L}_s$ of type (x), and show how it unfolds into a layer $\mathbf{L}_{s+1}$ of type (y), the transition being denoted (x)→(y). We denote by d_s the depth of the shallowest node in $\mathbf{L}_s$.

(i)→(i): The tree $q^{s+1}\mathcal{V}_k$ in each of the ℓ groups $\boldsymbol{\mathcal{M}}$ in $\mathbf{L}_s$ unfolds, by the definition of $\mathcal{V}_k$ (see also Figure 4), into a tree $q^{s+2}\mathcal{V}_k$ and $2^k - 1$ leaves of weight q^{s+1}, which provides a group $\boldsymbol{\mathcal{M}}$ for $\mathbf{L}_{s+1}$. Hence, there are ℓ groups $\boldsymbol{\mathcal{M}}$ in $\mathbf{L}_{s+1}$, which include $(2^k - 1)\ell$ signatures $s + 1$. This propagation of groups $\boldsymbol{\mathcal{M}}$ will occur in the same way in all the other transitions below; its discussion will be omitted for those cases. There remain $s + 2 - (2^k - 1)\ell = 2^{k-1} + 1 + j$ signatures $s + 1$, with $0 \leq j \leq 2^{k-1} - 4$ (recall that layers of type (i) exist only if $k > 2$). A quasi-uniform tree with $2^{k-1} + 2 + j$ leaves is built, rooted at $\mathcal{R}_s$. This tree has $2^{k-1} - (j+1) - 1$ leaves at depth $k - 1$, which are labeled $s + 1$, and $2(j + 1) + 2$ leaves at depth k, of which $2(j + 1) + 1$ are assigned label $s + 1$, and one serves as the root of $\mathcal{R}_{s+1}$, consistent with a structure of type (i) for $s + 1$ (and, correspondingly, $j + 1$).

(i)→(ii): We have $j = 2^{k-1} - 3$. We let $\mathcal{R}_s$ be the root of a balanced tree of height k. Of its 2^k leaves, $2^k - 2$ are assigned the remaining $2^k - 2$ signatures $s + 1$, one leaf serves as the root for $q\,\mathcal{U}_{k-1}$, and the remaining leaf as the root for $\mathcal{R}_{s+1}$.

(ii)→(iii) (k>2): The tree $q\,\mathcal{U}_{k-1}$ in $\mathbf{L}_s$ contributes 2^{k-1} leaves of signature $s + 1$ to $\mathbf{L}_{s+1}$, in addition to those contributed by the groups $\boldsymbol{\mathcal{M}}$. There remain $2^{k-1} - 1$ signatures $s + 1$, which are assigned to leaves of a balanced tree $\mathcal{U}_{k-1}$ rooted at $\mathcal{R}_s$. The remaining leaf splits into two nodes, one is the root of a tree $q\,\mathcal{U}_{k-1}$, and the other anchors $\mathcal{R}_{s+1}$.

(ii)→(iv) (k=2): The tree $q\,\mathcal{U}_1$ in $\mathbf{L}_s$ contributes 2^1 leaves of signature $s + 1$ to $\mathbf{L}_{s+1}$, in addition to those contributed by the groups $\boldsymbol{\mathcal{M}}$. The remaining signature $s + 1$ is assigned to one leaf of a tree $\mathcal{U}_1$ rooted at $\mathcal{R}_s$. The second leaf splits into two nodes, one is the root of a tree $q\mathcal{V}_k^-$, and the other anchors $\mathcal{R}_{s+1}$.

(iii)→(iii): The construction from the previous transition is kept, except that one of the leaves of the tree $\mathcal{U}_{k-1}$ rooted at $\mathcal{R}_s$ is split, making room for the additional signature $s+1$ resulting from the increase in s. Hence, there is a decrease by one in the number of leaves at depth d_s and an increase by two in the number of leaves at depth d_s+1. This process continues until $j = 2^k - 4$.

(iii)→(iv): This transition is identical to the previous one, except that instead of a tree $q\mathcal{U}_{k-1}$, a tree $q\mathcal{V}_k^-$ is attached as sibling to $\mathcal{R}_{s+1}$.

(iv)→(v): The tree $q\mathcal{V}_k^-$ from the previous transition provides the $2^{k-1}-1$ leaves of signature $s+1$, plus a tree $q\mathcal{V}_k$. What started as a balanced tree of depth $k-1$ in the transition (ii)→(iii) has evolved into a balanced tree of depth k, with all leaves assigned signatures $s+1$, except for one, which serves as the root of $\mathcal{R}_{s+1}$.

(v)→(i) (k>2): The tree $q\mathcal{V}_k$ added in the previous transition generates a new group $\boldsymbol{\mathcal{M}}$, consistent with the increment in ℓ. All signatures $s+1$ now originate from the groups $\boldsymbol{\mathcal{M}}$, or from $\mathcal{R}_s$, which brings the construction back to a layer of type (i), completing the cycle.

(v)→(ii) (k=2): When $k=2$ the transition occurs to a layer of type (ii), as described above for the initial transition from Case 1 to Case 2.

REFERENCES

[1] S. W. Golomb, "Run length encodings," *IEEE Trans. Inf. Theory*, vol. IT-12, pp. 399–401, 1966. 2, 3, 5, 10

[2] R. G. Gallager and D. C. Van Voorhis, "Optimal source codes for geometrically distributed integer alphabets," *IEEE Trans. Inf. Theory*, vol. IT-21, pp. 228–230, 1975. 2, 3, 4, 5, 6, 10, 12, 25, 27, 28

[3] N. Merhav, G. Seroussi, and M. J. Weinberger, "Optimal prefix codes for sources with two-sided geometric distributions," *IEEE Trans. Inf. Theory*, vol. 46, pp. 229–236, 2000. 2, 3, 6, 10, 27

[4] R. F. Rice, "Some practical universal noiseless coding techniques," Tech. Rep. JPL-79-22, JPL, Pasadena, CA, 1979. 2

[5] M. J. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS," *IEEE Trans. Image Proc.*, vol. 9, pp. 1309–1324, 2000. 2

[6] T. Linder, V. Tarokh, and K. Zeger, "Existence of optimal prefix codes for infinite source alphabets," *IEEE Trans. Inf. Theory*, vol. 43, pp. 2026–2028, 1997. 2

[7] A. Kato, T. S. Han, and H. Nagaoka, "Huffman coding with an infinite alphabet," *IEEE Trans. Inf. Theory*, vol. 42, pp. 977–984, 1996. 2

[8] J. Abrahams, "Code and parse trees for lossless source encoding," *Commun. Inf. Syst.*, vol. 1, pp. 113–146, 2001. 2

[9] M. J. Golin and K. K. Ma, "Algorithms for constructing infinite Huffman codes," Technical Report HKUST-TCSC-2004-07, HKUST, Hong Kong, China, July 2004. 2

[10] M. B. Baer, *Coding for General Penalties*. PhD thesis, Stanford University, 2003. 2, 3, 25

[11] S. W. Golomb, "Sources which maximize the choice of a Huffman coding tree," *Information and Control*, vol. 45, pp. 263–272, jun 1980. 3

[12] F. Bassino, M.-P. Béal, and D. Perrin, "A finite state version of the Kraft-McMillan theorem," *SIAM Journal on Computing*, vol. 30, no. 4, pp. 1211–1230, 2000. 3

[13] G. Seroussi and M. J. Weinberger, "On adaptive strategies for an extended family of Golomb-type codes," in *Proc. DCC'97*, (Snowbird, UT), pp. 131–140, 1997. 27